

Schmid – *Web Representation with Dynamic Thumbnails* In Combination**The following claims are invalid as being obvious over Schmid in view of the incorporated references**

Schmid, Stefan, *Web Representation with Dynamic Thumbnails*, IEEE YUFORICS '98 Conference, June 16-18, 1998.

As set forth below, Schmid et. al ('Schmid'), alone or in combination with each listed reference, discloses each and every element claimed in the listed claims of the '904 patent, when analyzed under the standard set forth in *KSR v. Teleflex*, 127, S. Ct. 1727 (2007):

- Exhibit 5: Cockburn, Greenberg et. al., ('Cockburn'), or
- Exhibit 6: CNN Interactive ('CNN Interactive'), or
- Exhibit 7: US Patent 6,356,908 ('Brown'), or
- Exhibit 8: Berners-Lee et. al. ('Berners-Lee'), or
- Exhibit 11: Akamai Freeflow ('Akamai'), or
- Exhibit 13: US Patent 6,594,697 ('Praitis') , or
- Exhibit 14: DoubleClick Ad Server ('Doubleclick'), or
- Exhibit 15: US Patent 6,108,703 ('Leighton'), or
- Exhibit 16: US Patent 5,761,436 ('Nielsen').

Claim #	The '904 Patent	Disclosure of Each Limitation in Schmid ¹
1.	A method for presenting Internet information to a user comprising:	"[W]e propose a novel online service to provide up-to-date thumbnails of any Web pages. Online provided and dynamically generated thumbnails open new ways to represent the Web and enhance Web designers potentialities." [Schmid, Abstract]
	providing to a user a visual image of a web page containing at least one hyperlink;	Schmid teaches an displaying a web page containing hyperlinks. "In order to enhance Web representation, we suggest to improve the power of expression of hyper-links. Text-only links, if well chosen, usually give information about the page content. Thumbnail links, where the image represents a miniature of the referenced

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

Web page, contain information of the page layout and the content. Combining both techniques the user may perceive a maximum of information regarding content and layout. See figure 1 as an example.” [Schmid at 1]

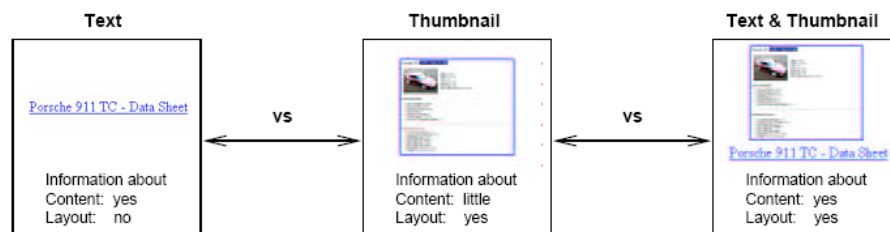


Figure 1: Textual links together with thumbnails contain information on both content and layout of linked pages

See also, Figure 2, which “shows how clients, the Thumbnail Server, the rendering engine, and Web servers or proxies are interrelated,” and includes a web browser with hyperlinks. [Schmid at 3.1]

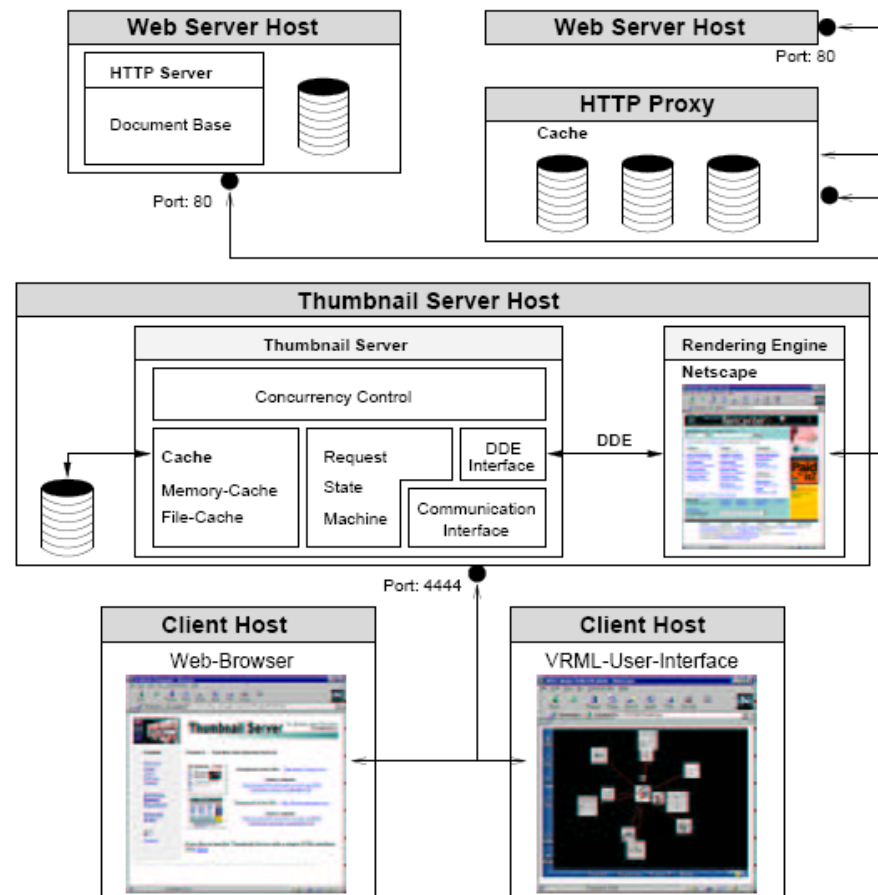

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

Figure 2: Thumbnail Server Architecture

The exact web page shown in the Web Browser within the Client Host in Figure 2 was also available on the World Wide Web; a screen snapshot of this page as retrieved via the Internet archive is shown below. I will refer to this image as ‘Snapshot 1’.

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

		 <p>In addition, in Figure 3, Schmid teaches displaying thumbnail visual images amongst other images and representations of hyperlinks, where thumbnails are 3-D objects in a three-dimensional space. .</p>
--	--	--

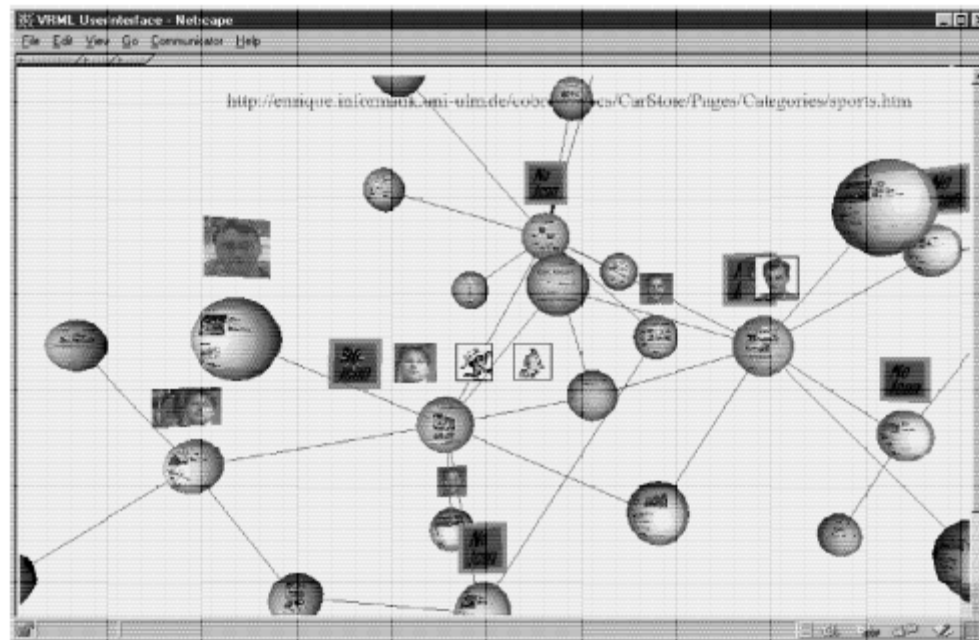
Schmid – *Web Representation with Dynamic Thumbnails In Combination*

Figure 3: CoBrow VRML User Interface

and at least partially concurrently providing a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image.

and at least partially concurrently providing: See Schmid at Fig. 1 and Fig. 2, *supra* (displaying thumbnail previews as in-line images that would load along with the web page).

a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink via the Internet:

Schmid teach the use of thumbnails of "any Web pages," which certainly includes homepages as a subset. Further, Schmid discloses sending a "request" URL from the client application to the Thumbnail Server that includes a URL variable: "The request format is designed to be HTTP compliant. Each request is encoded as an HTTP-GET [4] request. The general request syntax (in EBNF notation) is as follows: request ::= http://<server name>[:<port>]/?url=<url>{&<options>}.".

Schmid – *Web Representation with Dynamic Thumbnails In Combination*

		<p>Along with his accompanying discussion of URL parsing in Section 3.2, this suggests several methods that include displaying a home page. First, the EBNF notation specifies that the request comprises a url sent to the server specified in <server name>; this url would encompass URLs to a home page. An example of this is shown in Snapshot-1, <i>supra</i>. In this example, the second requested URL is home.netscape.com, clearly a page that Netscape considered to be its home page. The first requested URL is www.cobrow.com, a front page to the domain. Second, this protocol detaches the URL variable requested from the URL of the link the client browser is representing via a thumbnail, i.e., the client may request <i>any</i> URL from the thumbnail server. Consequently, one of ordinary skill in the art would have understood (for example) that the thumbnail could be the URL of a home page while the link was to an interior page.</p> <p><i>by employing an image server that stores and provides said thumbnail visual image:</i> Schmid discloses an image server and how it interacts with clients and other server. He calls a thumbnail server / thumbnail service. “The general architecture of the <i>Thumbnail Service</i> is based on a client-server model. Clients, such as ordinary Web browsers, Java applets, VRML programs, or other stand-alone programs, access the Thumbnail Server via the Internet protocol TCP/IP. A client requests thumbnails of arbitrary Web pages. After the request is received, the server first checks its memory and disk caches. If the requested Web page thumbnail is already rendered and not expired, it is immediately sent back to the client. Otherwise, the server loads the requested Web page and renders the image.” [Schmid at 3.1] See also Fig. 2, <i>supra</i> (illustrating a “Thumbnail Server Host” and a “Web Server Host”).</p> <p>If Schmid is found not to anticipate the ‘home page’ limitation, it would have been obvious to one of ordinary skill in the art to display a home page based on the disclosures in:</p> <ul style="list-style-type: none"> • Cockburn (Exhibit 5) <i>or</i> • CNN Interactive (Exhibit 6) <i>or</i> • Brown (Exhibit 7).
--	--	---

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

4.	A method according to claim 1 and wherein said thumbnail visual image is displayed within the visual image of said web page.	See Schmid at Fig. 2, <i>supra</i> , the client hosts at the bottom which show displaying the thumbnail image within the visual image of the web page. Also see
6.	A method according to claim 1 and wherein a plurality of thumbnail images represented by at least one hyperlink are displayed simultaneously along with said visual image of a web page containing at least one hyperlink.	As shown in Figures 2 and 3 <i>supra</i> , Schmid discloses a plurality of thumbnail images as described here.
7.	A method according to claim 1 and wherein said web page comprises an HTML page.	See Fig. 2, <i>supra</i> , showing the “Client Host” displaying an HTML page loaded in a “Web-Browser.” Schmid in Section 3.3 also states that his method uses external web browsers that obey the HTML standard: “No matter what new HTML features (e.g. frames, layers) are developed, we do not have to adjust our rendering engine.” Additionally, at the time of the alleged invention, web pages were almost always written in HTML.
12.	A method according to claim 1 and wherein said providing a-thumbnail visual image comprises: employing a web browser which interfaces via the Internet with a web server including visualization functionality.	See Fig. 2, <i>supra</i> , showing the “Client Host” displaying an HTML page loaded in a “Web-Browser.” Additionally: “The Thumbnail Server and the client applications use the Hyper Text Transfer Protocol (HTTP) as communication protocol. The server acts like a HTTP server regarding its clients.... The HTTP protocol allows users to request thumbnails directly from the Web browser according to RFC/1945 [4]. That offers the opportunity to simply integrate dynamically generated thumbnails into Web pages. Neither are specific client applications nor Web browser plug-ins necessary – merely a Web browser is requires as a client.” Schmid at 3.2.
13.	A method according to claim 12 and wherein said visualization	“An Application: The VRML User Interface of CoBrow. . . . The solution was to use a web of images consisting of user icons (representing neighbors) and Web page

Schmid – *Web Representation with Dynamic Thumbnails In Combination*

functionality is operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.

thumbnails (representing Web pages near the user's location) [3]. The need of a new service which provides dynamically generated thumbnails of arbitrary Web pages became obvious. Figure 3 presents a screen shot of the CoBrow VRML user interface using the Web page thumbnails."

See Figure 3 *supra*, a web browser parsing VRML commands to display web page thumbnails:

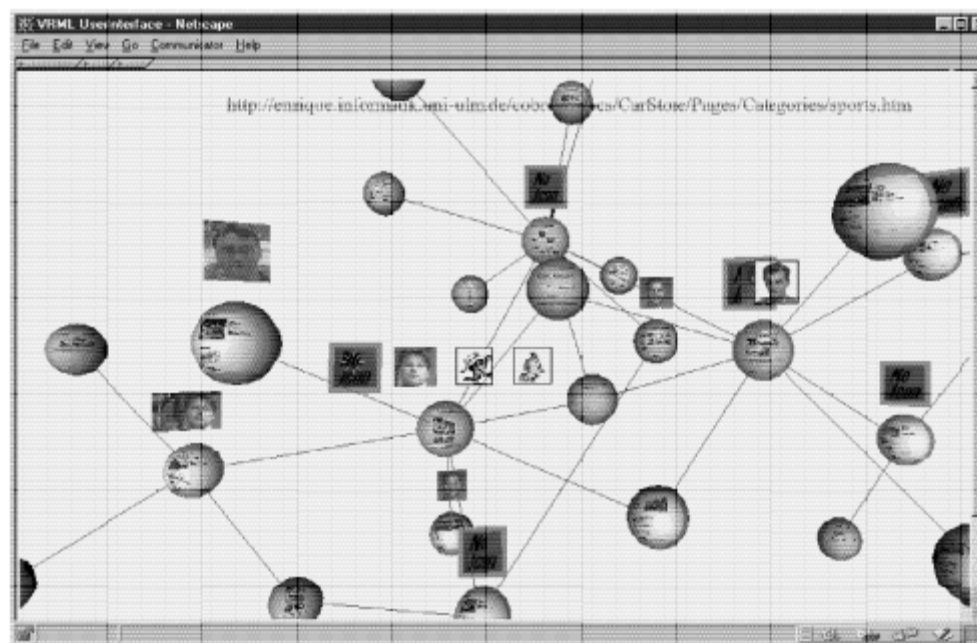


Figure 3: CoBrow VRML User Interface

14.

A method according to claim 13 and wherein said annotated web page includes the web page having within it thumbnail visual images of homepages of web sites

Schmid teach the use of thumbnails of "any Web pages," which certainly includes homepages as a subset.

As well, Schmid discloses sending a "request" URL from the client application to the Thumbnail Server that includes a URL variable: "The request format is designed

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

	referenced by hyperlinks contained in the web page.	<p>to be HTTP compliant. Each request is encoded as an HTTP-GET [4] request. The general request syntax (in EBNF notation) is as follows: request ::= http://<server name>[:<port>]/?url=<url>{&<options>}.’.</p> <p>Along with his accompanying discussion of URL parsing in Section 3.2, this suggests several methods that include displaying a home page. First, the EBNF notation specifies that the request comprises a url sent to the server specified in <server name>; this url would encompass URLs to a home page. Second, this protocol detaches the URL variable requested from the URL of the link the client browser is representing via a thumbnail, i.e., the client may request <i>any</i> URL from the thumbnail server. Consequently, one of ordinary skill in the art would have understood (for example) that the thumbnail could be the URL of a home page while the link was to an interior page.</p> <p>An example of this is shown in Snapshot-1, <i>supra</i>, which has requested thumbnails as designated by the URLs to two home pages: home.netscape.com, and www.cobrow.com.</p>
18.	A system for presenting Internet information to a user comprising:	See Claim 1 analysis, <i>supra</i> , which is hereby incorporated by reference.
	first functionality providing to a user a visual image of a web page containing at least one hyperlink; and	
	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and	



Schmid – *Web Representation with Dynamic Thumbnails* In Combination

	provides said thumbnail visual image.	
21.	A system according to claim 18 and wherein said thumbnail visual image is displayed within the visual image of said web page.	See Claim 4 analysis, <i>supra</i> , which is hereby incorporated by reference.
23.	A system according to claim 18 and wherein a plurality of thumbnail visual images represented by at least one hyperlink are displayed simultaneously along with said visual image of a web page containing at least one hyperlink.	See Claim 6 analysis, <i>supra</i> , which is hereby incorporated by reference.
24.	A system according to claim 18 and wherein said web page comprises an HTML page.	See Claim 7 analysis, <i>supra</i> , which is hereby incorporated by reference.
29.	A system according to claim 18 and wherein said second functionality comprises fourth functionality employing a web browser which interfaces via the Internet with a web server including visualization functionality.	See Claim 12 analysis, <i>supra</i> , which is hereby incorporated by reference.
30.	A system according to claim 29 and wherein said visualization functionality is operative to embed commands to the web browser to	See Claim 13 analysis, <i>supra</i> , which is hereby incorporated by reference.

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

	download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.	
31.	A system according to claim 30 and wherein said annotated web page includes the web page having within it thumbnail visual images of homepages of web sites referenced by hyperlinks contained in the web page.	See Claim 14 analysis, <i>supra</i> , which is hereby incorporated by reference.
35.	A method for presenting Internet information to a user comprising:	“[W]e propose a novel online service to provide up-to-date thumbnails of any Web pages. Online provided and dynamically generated thumbnails open new ways to represent the Web and enhance Web designers potentialities.” [Schmid, Abstract]
	providing to a user a visual image of a web page containing at least one hyperlink;	Schmid teaches displaying a web page containing hyperlinks. “In order to enhance Web representation, we suggest to improve the power of expression of hyper-links. Text-only links, if well chosen, usually give information about the page content. Thumbnail links, where the image represents a miniature of the referenced Web page, contain information of the page layout and the content. Combining both techniques the user may perceive a maximum of information regarding content and layout. See figure 1 as an example.” [Schmid at 1]

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

		<div><div><div><div>Text</div><div>Porsche 911 TC - Data Sheet</div><div>Information about Content: yes Layout: no</div></div><div>VS</div><div><div>Thumbnail</div><div></div><div>Information about Content: little Layout: yes</div></div><div>VS</div><div><div>Text & Thumbnail</div><div></div><div>Porsche 911 TC - Data Sheet Information about Content: yes Layout: yes</div></div></div><div><p>Figure 1: Textual links together with thumbnails contain information on both content and layout of linked pages</p><p>See also, Figure 2, which “shows how clients, the Thumbnail Server, the rendering engine, and Web servers or proxies are interrelated,” and includes a web browser with hyperlinks. [Schmid at 3.1]</p></div></div>
--	--	---

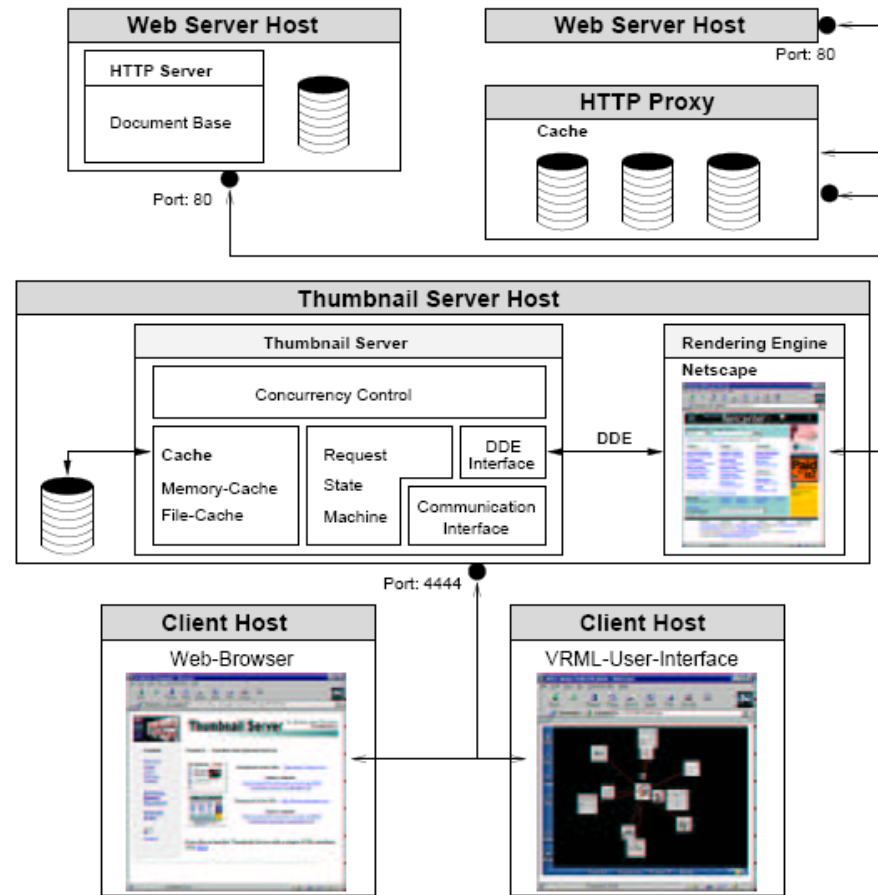
Schmid – *Web Representation with Dynamic Thumbnails In Combination*

Figure 2: Thumbnail Server Architecture

and at least partially concurrently providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one

and at least partially concurrently providing a thumbnail of another web page of at least one web site which is represented by said at least one hyperlink via the Internet: Schmid at Fig. 1 and Fig. 2, *supra* (displaying thumbnail previews as in-line images that would load along with the web page).

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

	<p>hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image,</p>	<p>Schmid teach the use of thumbnails of "any Web pages".</p> <p>As well, Schmid discloses sending a "request" URL from the client application to the Thumbnail Server that includes a URL variable: "The request format is designed to be HTTP compliant. Each request is encoded as an HTTP-GET [4] request. The general request syntax (in EBNF notation) is as follows: request ::= http://<server name>[:<port>]/?url=<url>{&<options>}.".</p> <p>Along with his accompanying discussion of URL parsing in Section 3.2, this suggests several methods that include displaying a 'another web page'. First, the EBNF notation specifies that the request comprises a url sent to the server specified in <server name>; this url would encompass URLs to a home page. Second, this protocol detaches the URL variable requested from the URL of the link the client browser is representing via a thumbnail, i.e., the client may request <i>any</i> URL from the thumbnail server. Consequently, one of ordinary skill in the art would have understood (for example) that the thumbnail could be the URL of 'another web page' while the link was to either the same page or a different page.</p> <p><i>by employing an image server that stores and provides said thumbnail visual image:</i> "The general architecture of the <i>Thumbnail Service</i> is based on a client-server model. Clients, such as ordinary Web browsers, Java applets, VRML programs, or other stand-alone programs, access the Thumbnail Server via the Internet protocol TCP/IP. A client requests thumbnails of arbitrary Web pages. After the request is received, the server first checks its memory and disk caches. If the requested Web page thumbnail is already rendered and not expired, it is immediately sent back to the client. Otherwise, the server loads the requested Web page and renders the image." [Schmid at 3.1] See also Fig. 2, <i>supra</i> (illustrating a "Thumbnail Server Host" separated from a "Web Server Host").</p>
	<p>said providing a thumbnail visual image comprising employing a web browser which interfaces via the Internet with a web server, separated from said image server, including visualization</p>	<p><i>said providing a thumbnail visual image comprising employing a web browser which interfaces via the Internet with a web server, separated from said image server:</i></p> <p>See Schmid at Fig. 2, <i>supra</i> (illustrating a "Thumbnail Server Host" separated from a "Web Server Host" and a "Web-Browser" displaying an annotated web page).</p>

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

	<p>functionality, said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.</p>	<p><i>including visualization functionality, said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page:</i></p> <p>“The Thumbnail Server and the client applications use the Hyper Text Transfer Protocol (HTTP) as communication protocol. The server acts like a HTTP server regarding its clients.... The HTTP protocol allows users to request thumbnails directly from the Web browser according to RFC/1945 [4]. That offers the opportunity to simply integrate dynamically generated thumbnails into Web pages. Neither are specific client applications nor Web browser plug-ins necessary – merely a Web browser is requires as a client.” Schmid at 3.2.</p> <p>If Schmid is found not to anticipate the ‘web server, separated from said image server’ limitation, it would have been obvious to one of ordinary skill in the art to employ a separate image server based on the disclosures in:</p> <ul style="list-style-type: none"> • Berners-Lee (Exhibit 8) <i>or</i> • Akamai (Exhibit 11), <i>or</i> • DoubleClick (Exhibit 14), <i>or</i> • Leighton (Exhibit 15).
38.	<p>A method according to claim 35 and wherein said thumbnail visual image is displayed within the visual image of said web page.</p>	<p>See Claim 4 analysis, <i>supra</i>, which is hereby incorporated by reference.</p>
40.	<p>A method according to claim 35 and wherein a plurality of thumbnail visual images represented by at least one hyperlink are displayed simultaneously along with said visual image of a web page</p>	<p>See Claim 6 analysis, <i>supra</i>, which is hereby incorporated by reference.</p>

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

	containing at least one hyperlink.	
41.	A method according to claim 35 and wherein said web page comprises an HTML page.	See Claim 7 analysis, <i>supra</i> , which is hereby incorporated by reference.
42.	A method according to claim 35 and wherein said annotated web page includes the web page having within it thumbnail visual images of homepages of web sites referenced by hyperlinks contained in the web page.	See Claim 13 analysis, <i>supra</i> , which is hereby incorporated by reference.
44.	A method according to claim 35 and wherein said visualization functionality comprises: receiving a list of hyperlinks;	<p>It would have been obvious to one of ordinary skill in the art to combine Schmid with the disclosures below to trim a URL, if desired:</p> <ul style="list-style-type: none"> • Praitis (Exhibit 13) • Nielsen (Exhibit 16).
	receiving a list of hyperlinks;	
	splitting a URL of each hyperlink into URL components including at least a path component and a host component;	
	trimming a path component based on the consideration of finding the most representative image of a given web page;	
	and constructing a new URL including a trimmed path component.	

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

46.	A system for presenting Internet information to a user comprising:	See Claim 35 analysis, <i>supra</i> , which is hereby incorporated by reference.
	first functionality providing to a user a visual image of a web page containing at least one hyperlink; and	
	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image, said second functionality comprising third functionality employing a web browser which interfaces via the Internet with a web server, separated from said image server, including visualization functionality,	
	said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to	

Schmid – *Web Representation with Dynamic Thumbnails In Combination*

	provide to a user, via the web browser, an annotated web page.	
49.	A system according to claim 46 and wherein said thumbnail visual image is displayed within the visual image of said web page.	See Claim 38 analysis, <i>supra</i> , which is hereby incorporated by reference.
50.	A system according to claim 49 and wherein said thumbnail visual image appears hovering over said hyperlink.	See Claim 39 analysis, <i>supra</i> , which is hereby incorporated by reference.
51.	A system according to claim 46 and wherein a plurality of thumbnail visual images represented by at least one hyperlink are displayed simultaneously along with said visual image of a web page containing at least one hyperlink.	See Claim 40 analysis, <i>supra</i> , which is hereby incorporated by reference.
52.	A system according to claim 46 and wherein said web page comprises an HTML page.	See Claim 41 analysis, <i>supra</i> , which is hereby incorporated by reference.
53.	A system according to claim 46 and wherein said annotated web page includes the web page having within it thumbnail visual images of homepages of web sites referenced by hyperlinks contained in the web page	See Claim 42 analysis, <i>supra</i> , which is hereby incorporated by reference.
55.	A system according to claim 46 and wherein said visualization	<i>See Claim 44 supra, which is hereby incorporated by reference.</i>

Schmid – *Web Representation with Dynamic Thumbnails* In Combination

	functionality comprises	
	receiving a list of hyperlinks;	
	splitting a URL of each hyperlink into URL components including at least a path component and a host component;	
	trimming a path component based on the consideration of finding the most representative image of a given web page; and	
	constructing a new URL including a trimmed path component.	

EXHIBIT G4

Kopetzky et al., *Visual preview for link traversal on the World Wide Web***The following claims are invalid as being obvious over Kopetsky in view of the incorporated references**

Kopetzky, T. and Muhlhauser, M., *Visual preview for link traversal on the World Wide Web*. Computer Networks Vol. 31, No. 11 (May 17, 1999) pages 1525-1532.


As set forth below, Kopetsky, et. al. ('Kopetzky'), alone or in combination with each listed reference, discloses each and every element claimed in the listed claims of the '904 patent, when analyzed under the standard set forth in *KSR v. Teleflex*, 127, S. Ct. 1727 (2007):

- Exhibit 5: Cockburn, Greenberg et. al., ('Cockburn'), or
- Exhibit 6: CNN Interactive ('CNN Interactive'), or
- Exhibit 7: US Patent 6,356,908 ('Brown'), or
- Exhibit 8: Berners-Lee et. al. ('Berners-Lee'), or
- Exhibit 9: 6,058,417 (Hess), or
- Exhibit 10: Frankel, Swain et. al., ('Frankel'), or
- Exhibit 11: Akamai Freeflow ('Akamai'), or
- Exhibit 12: Sclaroff et. al. ('Sclaroff') or
- Exhibit 13: US Patent 6,594,697 ('Praitis') , or
- Exhibit 14: DoubleClick Ad Server ('Doubleclick'), or
- Exhibit 15: US Patent 6,108,703 ('Leighton'), or
- Exhibit 16: US Patent 5,761,436 ('Nielsen').

Claim #	The '904 Patent	Disclosure of Each Limitation in Kopetsky ¹
1.	A method for presenting Internet information to a user comprising:	Kopetzky discloses a "visual link preview" to present Internet information for view by end users for the purpose of helping them make a "decision about which links to follow and which to ignore." [Kopetsky, page 1525]


¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

	providing to a user a visual image of a web page containing at least one hyperlink; and	A visual image of a webpage containing hyperlinks is displayed to a user via the web browser. <i>See e.g.</i> , pp. 1526-1527.
	at least partially concurrently providing a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image.	<p>Kopetzky discloses providing a thumbnail preview of the webpage (in this case a home page) associated with the hyperlink. <i>See e.g.</i>, pp. 1527. The thumbnail image is a small JPEG image file, where Fig. 2 (reproduced below) on page 1527 illustrates two JPEG thumbnails produced by two different methods. The thumbnail is of the Telecooperation Department home page illustrated in Fig. 1 on page 1526.</p>  <p>(a) (b)</p> <p>Fig. 2. From [Kopetsky, page 1527]</p> <p>Kopetsky discloses that the thumbnail image is to a home page. “The following list shows which link types are recognized by our system and how they are visualized. As the linking mechanism works with URLs we are using properties of the URLs to categorize a link.</p> <ul style="list-style-type: none"> • The URL points to the beginning of a Web page, as in http://www.tk.unilinz.ac.at/. Links of this type are visualized using a thumbnail picture as in Fig. 2b.” [Kopetsky, Page 1527, emphasis added] <p>Kopetsky also discloses that the thumbnail image may represent a location other than the one pointed to in the link. In the example below, the thumbnail would be to the URL trimmed to exclude the #hypertext anchor, although the link associated with that thumbnail would be to the URL including the anchor.</p> <p>“The URL points to an anchor, as in http://www.encyclopedia.com/h.html#hypertext. Links of this type may be visualized using a thumbnail picture or, if there is text after</p>

		<p>the anchor, the text referenced by the link itself may be displayed (see Fig. 3a).” [Kopetsky, Page 1527, emphasis added]</p> <p>A hypertext “anchor,” indicated by the “#” sign, creates an interior link on a web document, so that a user can navigate to different locations within the same web page or document. In Kopetsky’s example, the www.encyclopedia.com/h.html page is likely an alphabetical list of all encyclopedia entries starting with the letter H. The anchor is to “hypertext,” an entry that would be near the bottom of the “h.html” page.</p> <p>Kopetsky discloses a proxy server / cache that functions as an image server that stores and provides the thumbnail visual image. <i>See e.g.</i>, pp. 1529, 1530 for detail. Kopetzky introduces the proxy server as follows:</p> <p>“To solve this problem an approach using a proxy server was chosen. An overview of the main components of the proxy server, which has been implemented in Java, can be seen in Figure 7.</p> <p>The proxy server has the following tasks:</p> <ul style="list-style-type: none"> • analyze the links in the requested HTML document and generate the preview images for all links in the document; • cache the requested HTML documents and the computed link preview images for future access; • modify the HTML documents in a way that the requesting browser is able to show the link preview images. <p>This approach has the following advantages:</p> <ul style="list-style-type: none"> • the proxy server has to generate the preview information only once (depending on server space); • many readers can share one proxy server and thus benefit from already generated preview information; • readers only have to make one change in their browsing environment: they have to configure the Web client to use a proxy server – everything else is done automatically; • the proxy server can use other proxies servers and thus benefit from information already fetched from the Web.” [Kopetsky, pages 1528-1529]
--	--	---

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

		<p>The text following the above description, although not repeated here, clarifies that the proxy server is separate from both the browser asking for a requested document and from other servers that deliver the source document, i.e., as with most proxy servers, it acts as a separate intermediary between the two.</p> <p>Kopetzky also expressly discloses modifying “Link services [that] store link information external to the document linked” so that it can store thumbnail information as well as the link information. This modification defines a separate image server. [Kopetsky, Section 6.4 page 1531]</p> <p>If Kopetzky is found not to anticipate the ‘home page’ limitation, it would have been obvious to one of ordinary skill in the art to display a home page based on the disclosures in:</p> <ul style="list-style-type: none"> • Cockburn (Exhibit 5) <i>or</i> • CNN Interactive (Exhibit 6) <i>or</i> • Brown (Exhibit 7).
4	A method according to claim 1 and wherein said thumbnail visual image is displayed within the visual image of said web page.	<p>Kopetzky discloses providing the thumbnail preview within the visual image of the webpage. <i>See e.g.</i>, 1527-1528.</p> <p>Figure 5 from Kopetsky, reproduced below, illustrates the thumbnail preview within the visual image of the page.</p> 
5.	A method according to claim 4 and wherein said thumbnail visual	Kopetzky discloses providing the thumbnail preview upon a mouse-over event over a link. e.g.,

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

	image appears hovering over said hyperlink.	“The link preview is simply activated by moving the mouse over a link. The presentation of the link preview is animated. This means that the reader has time to accommodate to the new situation. The preview opens below the link and will remain open for seven seconds. The preview can be closed by moving the mouse over the preview and then out of it. The preview will close also when the reader moves the mouse over another link and thus activating a new preview image.” [Kopetsky, page 1528]
6.	A method according to claim 1 and wherein a plurality of thumbnail visual images represented by at least one hyperlink are displayed simultaneously along with said visual image of a web page containing at least one hyperlink.	<p>Kopetzky discloses applying thumbnail preview images for every hyperlink on a web page. <i>See e.g.</i>, pp. 1527-1529 and example below.</p> <p>“The link preview is simply activated by moving the mouse over a link. The presentation of the link preview is animated. This means that the reader has time to accommodate to the new situation. The preview opens below the link and will remain open for seven seconds. The preview can be closed by moving the mouse over the preview and then out of it. The preview will close also when the reader moves the mouse over another link and thus activating a new preview image.” [Kopetsky, page 1528]</p> <p>Kopetsky also discloses that more control over the preview can be provided, e.g., to have the preview always on to provide a plurality of thumbnail images: “More control over the preview. Currently, control is limited to the states “preview on” and “preview off”. Users could, for example, define which link to preview in which manner.” [Kopetsky, page 1531]</p>
7.	A method according to claim 1 and wherein said web page comprises an HTML page.	Kopetzky discloses that the web page comprises an HTML page. <i>See e.g.</i> , pp. 1528-1529. For example, “the proxy server has the following tasks: analyze the links in a requested HTML document...” (Kopetsky, page 1528)
12.	A method according to claim 1 and wherein said providing a thumbnail visual image comprises: employing a web browser which	<p>Kopetzky discloses employing a web browser that interfaces using standard HTTP protocols over the Internet with a proxy server that functions as a web server. The proxy server (web server) includes visualization functionality that modifies the HTML code for the web page to include Javascript code that enables link preview to provide a thumbnail visual image. <i>See e.g.</i>, pp. 1528-1530.</p> <p>Kopetzky in Section 4 describes the client-side implementation of the web browser,</p>

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

	interfaces via the Internet with a web server including visualization functionality.	while Section 3 describes the architecture of the proxy server, as well as the interplay between the two.
13.	A method according to claim 12 and wherein said visualization functionality is operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.	<p>Kopetzky discloses visualization functionality that embeds Javascript code into the HTML code for the webpage. <i>See e.g.</i>, pp. 1528-1530. The Javascript code commands the web browser to download, via the proxy server that functions as an image server, thumbnail visual images of webpages which represent hyperlinks contained in the webpage. <i>See e.g.</i>, pp. 1528-1530. A resulting annotated webpage including the thumbnail preview image is provided to a user via the web browser.</p> <p>Kopetzky Section 4.1 provides details, an extract which is reproduced below. “To enable the browser to show the link preview, the proxy server inserts a layer definition for invisible layers and a short JavaScript program at the beginning of each downloaded HTML-page. An example for a layer definition can be seen in Fig. 8. The position where the insertion actually takes place depends on the structure of the document (if it contains a head tag or a frame set, for example). Additionally each <A HREF> tag is modified to react to events when the mouse pointer moves over the link. A mouseover event-handler is inserted into each link description. This handler will activate a procedure in the inserted JavaScript program if the reader moves his mouse over the link. The handler passes as a parameter the name of the preview image to the procedure. This name was inserted into the link definition just as the event-handler. The procedure will make the invisible layer visible and set some parameter of the layer, e.g. the information which preview image is to be shown.” [Kopetzky, page 1529]</p>
14.	A method according to claim 13 and wherein said annotated web page includes the web page having within it thumbnail visual images of homepages of web sites referenced by hyperlinks contained in the web page.	<p>Kopetzky discloses providing a thumbnail preview of the webpage (in this case a home page) associated with the hyperlink. <i>See e.g.</i>, pp. 1527. The thumbnail image is a small JPEG image file, where Fig. 2, <i>supra</i>, on page 1527 illustrates two JPG thumbnails produced by two different methods. The thumbnail is of the Telecooperation Department home page illustrated in Fig. 1, <i>supra</i>, on page 1526.</p> <p>Kopetsky disclose that the thumbnail image is to a home page. “The following list shows which link types are recognized by our system and how they are visualized. As the linking mechanism works with URLs we are using properties of the URLs to</p>

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

		<p>categorize a link.</p> <ul style="list-style-type: none"> • The URL points to the beginning of a Web page, as in http://www.tk.uni-linz.ac.at/. Links of this type are visualized using a thumbnail picture as in Fig. 2b.” [Kopetsky, Page 1527, emphasis added] <p>Kopetsky also discloses that more control over the preview can be provided, e.g., to have the preview always on to provide a plurality of thumbnail images: “More control over the preview. Currently, control is limited to the states “preview on” and “preview off”. Users could, for example, define which link to preview in which manner.” [Kopetsky, page 1531]</p> <p>Kopetsky discloses that these homepages of web sites are referenced by hyperlinks: “To follow the previewed link, the user has the possibility to activate the link itself as usual or to click on the preview image.” [Kopetsky, page 1528]</p>
15.	A method according to claim 1 and wherein said thumbnail visual image appears hovering over said hyperlink.	See Claim 5 analysis, <i>supra</i> , which is hereby incorporated by reference.
16.	A method for generating an image server database of thumbnail visual images of web pages, the method comprising:	Kopetzky discloses generating and storing thumbnail visual images of web pages.
	receiving a list of URLs corresponding to said web pages, the thumbnail visual images of which it is desired to supply to said image server database;	<p>In his discussion of the architecture of the proxy server, Kopetzky discloses receiving a list of URLs corresponding to desired thumbnail visual images:</p> <p>“The proxy server has the following tasks:</p> <ul style="list-style-type: none"> • analyze the links in a requested HTML document and generate the preview images for all links in the document;” [Kopetzky, page 1528, emphasis added]
	operating a multiplicity of downloaders simultaneously to retrieve from the Internet, web pages and embedded objects	<p>For each link, an internal web browser is started to retrieve, in parallel simultaneously from the Internet, web pages and embedded objects corresponding to URLs from the list:</p> <p>“(4) The parser analyses the document regarding its structure and searches for link</p>

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

	corresponding to URLs from said list;	<p>information. If link information is found, for each link an internal Web browser will be started. These browsers will be used to generate the preview images.” [Kopetzky, page 1529, emphasis added]</p> <p>In addition, when describing the activities of the proxy server, Kopetzky discloses that web pages are retrieved / downloaded simultaneously (in parallel) to speed up processing: “Part of these activities happen in parallel to speed up processing.” [Kopetzky, page 1529]</p>
	causing a thumbnail generator to render retrieved web pages retrieved simultaneously by said multiplicity of downloaders; and causing said thumbnail generator to shrink said rendered images of said retrieved web pages and supply them to said image server database.	<p>The thumbnail generator renders and shrinks the images of the retrieved web pages and supplies them to the proxy server cache for later use: “The proxy server has the following tasks:</p> <ul style="list-style-type: none"> • analyze the links in a requested HTML document and generate the preview images for all links in the document; • cache the requested HTML documents and the computed link preview images for future access;” [Kopetzky, page 1528-1529, emphasis added] <p>One skilled in the art would know that storing the computed link preview images in a cache could be done with a database. Kopetzky acknowledges this when he expressly discloses in his Section 6.4 modifying “Link services [that] store link information external to the document linked” so that it can store thumbnail information as well as the link information. In this disclosure, he specifically references a link service by L.A. Carr, D. De Roure, W. Hall and G. Hill titled <i>The Distributed Link Service: A Tool for Publishers, Authors, and Readers</i>, published in: Proc. 4th Int. World Wide Web Conference, Boston, MA, USA, O’Reilly and Associates, December, 1995. In that reference, the link service stores its information in one or more databases called link databases.</p> <p>When describing the activities of the proxy server, Kopetzky discloses that web pages are retrieved / downloaded simultaneously (in parallel) to speed up processing. For each link, an internal web browser is started to retrieve, in parallel simultaneously from the Internet, web pages and embedded objects corresponding to URLs from the list:</p>

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

		<p>“(4) The parser analyses the document regarding its structure and searches for link information. If link information is found, for each link an internal Web browser will be started. These browsers will be used to generate the preview images.” [Kopetzky, page 1529]</p> <p>Kopetzky adds: “Part of these activities happen in parallel to speed up processing.” [Kopetzky, page 1529]</p> <p>If Kopetzky is found not to anticipate the ‘multiplicity of downloaders’ limitation or the ‘image server database’ limitation, it would have been obvious to one of ordinary skill in the art to employ the ‘multiplicity of downloaders’ and/or an ‘image server database’ based on the disclosures in:</p> <ul style="list-style-type: none"> • Hess (Exhibit 9) <i>or</i> • Frankel (Exhibit 10) <i>or</i> • Sclaroff (Exhibit 12).
17.	A method according to claim 16 also comprising deleting executable content from said retrieved web pages.	<p>The thumbnail generated image of the retrieved web page necessarily deletes executable content, as images do not contain executable content.</p> <p>If Kopetzky is found not to anticipate the ‘deleting executable content’ limitation, it would have been obvious to one of ordinary skill in the art to delete executable content based on the disclosures in:</p> <ul style="list-style-type: none"> • Hess (Exhibit 9) <i>or</i> • Frankel (Exhibit 10) <i>or</i> • Sclaroff (Exhibit 12).
18.	A system for presenting Internet information to a user comprising:	See Claim 1 analysis, <i>supra</i> , which is hereby incorporated by reference.
	first functionality providing to a user a visual image of a web page	

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

	containing at least one hyperlink; and	
	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image.	
21.	A system according to claim 18 and wherein said thumbnail visual image is displayed within the visual image of said web page.	See Claim 4 analysis, <i>supra</i> , which is hereby incorporated by reference.
22.	A system according to claim 21 and wherein said thumbnail visual image appears hovering over said hyperlink.	See Claim 5 analysis, <i>supra</i> , which is hereby incorporated by reference.
23.	A system according to claim 18 and wherein a plurality of thumbnail visual images represented by at least one hyperlink are displayed simultaneously along with said visual image of a web page containing at least one hyperlink.	See Claim 6 analysis, <i>supra</i> , which is hereby incorporated by reference.
24.	A system according to claim 18 and wherein said web page comprises an HTML page.	See Claim 7 analysis, <i>supra</i> , which is hereby incorporated by reference.

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

29.	A system according to claim 18 and wherein said second functionality comprises fourth functionality employing a web browser which interfaces via the Internet with a web server including visualization functionality.	See Claim 12 analysis, <i>supra</i> , which is hereby incorporated by reference.
30.	A system according to claim 29 and wherein said visualization functionality is operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.	See Claim 13 analysis, <i>supra</i> , which is hereby incorporated by reference.
31.	A system according to claim 30 and wherein said annotated web page includes the web page having within it thumbnail visual images of homepages of web sites referenced by hyperlinks contained in the web page.	See Claim 14 analysis, <i>supra</i> , which is hereby incorporated by reference.
32.	A system according to claim 18 and wherein said thumbnail visual image appears hovering over said hyperlink.	See Claim 15 analysis, <i>supra</i> , which is hereby incorporated by reference.
33.	A system for generating an image server database of thumbnail visual images of web pages, the	See 16 analysis, <i>supra</i> , which is hereby incorporated by reference.

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

	system comprising:	
	a multiplicity of downloaders, each receiving at least one URL from a list of URLs corresponding to said web pages, the thumbnail visual images of which it is desired to supply to said image server database, and	
	simultaneously retrieving from the Internet web pages and embedded objects corresponding to said at least one URL; and	
	at least one thumbnail generator operative to render the web pages, shrink said rendered images of the web pages and supply said rendered images to said image server database.	
34.	A system according to claim 33 and wherein said multiplicity of downloaders are operative to delete executable content from the web pages.	See 17 analysis, <i>supra</i> , which is hereby incorporated by reference.
35.	A method for presenting Internet information to a user comprising:	Kopetzky discloses a “visual link preview” to present Internet information for view by end users for the purpose of helping them make a “decision about which links to follow and which to ignore” [Kopetsky, page 1525].
	providing to a user a visual image of a web page containing at least one hyperlink; and	A visual image of a webpage containing hyperlinks is displayed to a user via the web browser. <i>See e.g.</i> , pp. 1526-1527.
	at least partially concurrently providing a thumbnail visual image of another web page of at least one web site which is	Kopetzky discloses providing a thumbnail preview of the webpage (in this case a home page) associated with the hyperlink. <i>See e.g.</i> , p. 1527. The thumbnail image is a small JPEG image file, where Fig. 2 (reproduced below) on page 1527 illustrates two JPEG thumbnails produced by two different methods. The thumbnail is of the

represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image,

Telecooperation Department home page illustrated in Fig. 1 on page 1526.

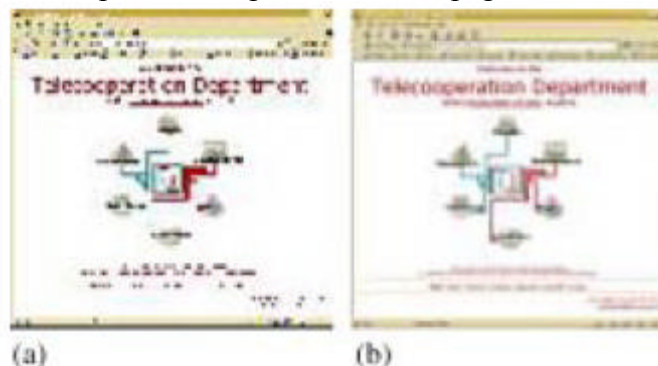


Fig. 2. From [Kopetsky, page 1527]

Kopetsky discloses that the thumbnail image is to a home page. “The following list shows which link types are recognized by our system and how they are visualized. As the linking mechanism works with URLs we are using properties of the URLs to categorize a link.

- **The URL points to the beginning of a Web page, as in <http://www.tk.unilinz.ac.at/>. Links of this type are visualized using a thumbnail picture as in Fig. 2b.” [Kopetsky, Page 1527, emphasis added]**

Kopetsky also disclose that the thumbnail image may represent a page other than the one pointed to in the link. In the example below, the thumbnail would be to the URL trimmed to exclude the #hypertext anchor, although the link associated with that thumbnail would be to the URL including the anchor.

“ The URL points to an anchor, as in <http://www.encyclopedia.com/h.html#hypertext>. Links of this type may be visualized using a thumbnail picture or, if there is text after the anchor, the text referenced by the link itself may be displayed (see Fig. 3a).” [Kopetsky, Page 1527, emphasis added]

A hypertext “anchor,” indicated by the “#” sign, creates an interior link on a web document, so that a user can navigate to different locations within the same web page or document. In Kopetsky’s example, the www.encyclopedia.com/h.html page is likely an alphabetical list of all encyclopedia entries starting with the letter H. The

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

		anchor is to “hypertext,” an entry that would be near the bottom of the “h.html” page.
	said providing a thumbnail visual image comprising employing a web browser which interfaces via the Internet with a web server, separated from said image server, including visualization functionality,	<p>Kopetsky discloses a separate proxy server / cache that functions as an image server that stores and provides the thumbnail visual image. <i>See e.g.</i>, pp. 1529, 1530 for detail. Kopetzky introduces the proxy server as follows:</p> <p>“To solve this problem an approach using a proxy server was chosen. An overview of the main components of the proxy server, which has been implemented in Java, can be seen in Figure 7.</p> <p>The proxy server has the following tasks:</p> <ul style="list-style-type: none"> • analyze the links in the requested HTML document and generate the preview images for all links in the document; • cache the requested HTML documents and the computed link preview images for future access; • modify the HTML documents in a way that the requesting browser is able to show the link preview images. <p>This approach has the following advantages:</p> <ul style="list-style-type: none"> • the proxy server has to generate the preview information only once (depending on server space); • many readers can share one proxy server and thus benefit from already generated preview information; • readers only have to make one change in their browsing environment: they have to configure the Web client to use a proxy server – everything else is done automatically; • the proxy server can use other proxies servers and thus benefit from information already fetched from the Web.” [Kopetsky, pages 1528-1529] <p>The text following the above description, although not repeated here, clarifies that the proxy server is separate from both the browser asking for a requested document and from other servers that deliver the source document, i.e., as with most proxy servers, it acts as a separate intermediary between the two.</p> <p>Kopetzky also expressly discloses modifying “Link services [that] store link</p>

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

		information external to the document linked” so that it can store thumbnail information as well as the link information. This modification defines a separate image server. [Kopetsky, Section 6.4 page 1531]
	said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.	<p>Kopetzky discloses visualization functionality that embeds Javascript code into the HTML code for the webpage. <i>See e.g.</i>, pp. 1528-1530. The Javascript code commands the web browser to download, via the proxy server that functions as an image server, thumbnail visual images of webpages which represent hyperlinks contained in the webpage. <i>See e.g.</i>, pp. 1528-1530. A resulting annotated webpage including the thumbnail preview image is provided to a user via the web browser.</p> <p>Kopetzky Section 4.1 provides details, an extract which is reproduced below. “To enable the browser to show the link preview, the proxy server inserts a layer definition for invisible layers and a short JavaScript program at the beginning of each downloaded HTML-page. An example for a layer definition can be seen in Fig. 8. The position where the insertion actually takes place depends on the structure of the document (if it contains a head tag or a frame set, for example). Additionally each <A HREF> tag is modified to react to events when the mouse pointer moves over the link. A mouseover event-handler is inserted into each link description. This handler will activate a procedure in the inserted JavaScript program if the reader moves his mouse over the link. The handler passes as a parameter the name of the preview image to the procedure. This name was inserted into the link definition just as the event-handler. The procedure will make the invisible layer visible and set some parameter of the layer, e.g. the information which preview image is to be shown.” [Kopetzky, page 1529]</p> <p>If Kopetzky is found not to anticipate the ‘web server, separated from said image server’ limitation, it would have been obvious to one of ordinary skill in the art to employ a separate image server based on the disclosures in:</p> <ul style="list-style-type: none"> • Berners-Lee (Exhibit 8) <i>or</i> • Akamai (Exhibit 11), <i>or</i> • DoubleClick (Exhibit 14), <i>or</i> • Leighton (Exhibit 15).

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

38.	A method according to claim 35 and wherein said thumbnail visual image is displayed within the visual image of said web page.	See Claim 4 analysis, <i>supra</i> , which is hereby incorporated by reference.
39.	A method according to claim 38 and wherein said thumbnail visual image appears hovering over said hyperlink. containing at least one hyperlink.	See Claim 5 analysis, <i>supra</i> , which is hereby incorporated by reference.
40.	A method according to claim 35 and wherein a plurality of thumbnail visual images represented by at least one hyperlink are displayed simultaneously along with said visual image of a web page containing at least one hyperlink.	See Claim 6 analysis, <i>supra</i> , which is hereby incorporated by reference.
41.	A method according to claim 35 and wherein said web page comprises an HTML page.	See Claim 7 analysis, <i>supra</i> , which is hereby incorporated by reference.
42.	A method according to claim 35 and wherein said annotated web page includes the web page having within it thumbnail visual images of homepages of web sites referenced by hyperlinks contained in the web page.	See Claim 13 analysis, <i>supra</i> , which is hereby incorporated by reference.
44.	A method according to claim 35 and wherein said visualization functionality comprises: receiving a list of hyperlinks;	It would have been obvious to one of ordinary skill in the art to combine Kopetzky with the disclosures below to trim a URL, if desired:

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

	receiving a list of hyperlinks;	<ul style="list-style-type: none"> • Praitis (Exhibit 13) • Nielsen (Exhibit 16).
	splitting a URL of each hyperlink into URL components including at least a path component and a host component;	
	trimming a path component based on the consideration of finding the most representative image of a given web page;	
	and constructing a new URL including a trimmed path component.	
45.	A method according to claim 35 and wherein said thumbnail visual image appears hovering over said hyperlink.	See Claim 15 analysis, <i>supra</i> , which is hereby incorporated by reference.
46.	A system for presenting Internet information to a user comprising:	See Claim 35 analysis, <i>supra</i> , which is hereby incorporated by reference.
	first functionality providing to a user a visual image of a web page containing at least one hyperlink; and	A visual image of a webpage containing hyperlinks is displayed to a user via the web browser. <i>See e.g.</i> , pp. 1526-1527.
	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one hyperlink via the Internet by	Kopetzky discloses providing a thumbnail preview of the webpage associated with the hyperlink. <i>See e.g.</i> , pp. 1527. The thumbnail image is a small JPEG image file. <i>See e.g.</i> , pp. 1527; <i>Compare</i> Fig. 1 and Fig. 2. Kopetzky further discloses at least partially concurrently providing the thumbnail preview with the visual image of the web page. <i>See e.g.</i> , pp. 1527. Kopetzky also discloses alternatively providing the thumbnail upon a mouse-over event over a link. <i>See e.g.</i> , pp. 1528. A proxy server that functions as an image server stores and provides the thumbnail visual image. <i>See e.g.</i> , pp. 1529, 1530.

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

	employing an image server that stores and provides said thumbnail visual image,	Kopetzky also expressly discloses modifying the server disclosed in <i>The Distributed Link Service: A Tool for Publishers, Authors, and Readers</i> , Proc. 4th Int. World Wide Web Conference, Boston, Mass., December 1995 (L.A. Carr), to store and provide images as part of the link service.
	said second functionality comprising third functionality employing a web browser which interfaces via the Internet with a web server, separated from said image server, including visualization functionality,	Kopetzky discloses employing a web browser that interfaces using standard HTTP protocols over the Internet with a proxy server that functions as a web server. The same proxy server separately functions as both a web server and an image server. The proxy server (web server) includes visualization functionality that modifies the HTML code for the web page to include Javascript code that enables link preview to provide a thumbnail visual image. <i>See e.g.</i> , pp. 1528-1530.
	said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.	Kopetzky discloses visualization functionality that embeds Javascript code into the HTML code for the webpage. <i>See e.g.</i> , pp. 1528-1530. The Javascript code commands the web browser to download, via the proxy server that functions as an image server, thumbnail visual images of webpages which represent hyperlinks contained in the webpage. <i>See e.g.</i> , pp. 1528-1530. A resulting annotated webpage including the thumbnail preview image is provided to a user via the web browser.
49.	A system according to claim 46 and wherein said thumbnail visual image is displayed within the visual image of said web page.	See Claim 38 analysis, <i>supra</i> , which is hereby incorporated by reference.
50.	A system according to claim 49 and wherein said thumbnail visual image appears hovering over said hyperlink.	See Claim 39 analysis, <i>supra</i> , which is hereby incorporated by reference.
51.	A system according to claim 46 and wherein a plurality of thumbnail visual images represented by at least one	See Claim 40 analysis, <i>supra</i> , which is hereby incorporated by reference.

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

	hyperlink are displayed simultaneously along with said visual image of a web page containing at least one hyperlink.	
52.	A system according to claim 46 and wherein said web page comprises an HTML page.	See Claim 41 analysis, <i>supra</i> , which is hereby incorporated by reference.
53.	A system according to claim 46 and wherein said annotated web page includes the web page having within it thumbnail visual images of homepages of web sites referenced by hyperlinks contained in the web page.	See Claim 42 analysis, <i>supra</i> , which is hereby incorporated by reference.
55.	A system according to claim 46 and wherein said visualization functionality comprises	<i>See Claim 44 supra, which is hereby incorporated by reference.</i>
	receiving a list of hyperlinks;	
	splitting a URL of each hyperlink into URL components including at least a path component and a host component;	
	trimming a path component based on the consideration of finding the most representative image of a given web page; and	
	constructing a new URL including a trimmed path component.	
56.	A system according to claim 46	See Claim 45 analysis, <i>supra</i> , which is hereby incorporated by reference.

Kopetzky et al., *Visual preview for link traversal on the World Wide Web*

	and wherein said thumbnail visual image appears hovering over said hyperlink.	
--	---	--

EXHIBIT G5

Cockburn: Webview

**The following claims are invalid as being obvious under Cockburn in view of
Brown, or Kraft, or Schmid, or Kopetzky**

Cockburn, A., Greenberg, S., McKenzie, B., Jasonsmith, M. and Kaasten, S., *WebView: A Graphical Aid for Revisiting Web Pages*. In Proceedings of the Australian Conference on Human Computer Interaction - OZCHI'99. (Wagga Wagga, Australia), pages 15-22, November 28-30, 1999.

I note that I am a co-author of this paper. I also note that this is a different paper than the *Issues in page representation and organization in web browser's revisitation tools* article by Cockburn presented at the same conference in the same year and referenced on the face of the '904 patent. This paper refers to the other paper as a companion paper, thus showing its close linkage to it.

Claim #	The '904 Patent	Disclosure of Each Limitation in Cockburn ¹
1.	A method for presenting Internet information to a user comprising:	
	providing to a user a visual image of a web page containing at least one hyperlink;	<i>Webview collects thumbnail visual images of web pages, and provides these to the user as a visual list of web page thumbnails alongside a web browser.</i> "WebView is an add-on window that interacts with unaltered versions of Netscape Navigator1. Whenever the user visits a page in Netscape, WebView's display is automatically updated to reflect the action." [Cockburn, Section 3]
	and at least partially concurrently providing a thumbnail visual image of the home page of at least one web site which is represented	<u><i>Thumbnail visual image of a home page.</i></u> <i>Webview provides thumbnail visual images, including thumbnail visual images of home pages.</i>

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

Cockburn: Webview

by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image.

“3.1 Representing pages. WebView captures and displays a miniaturised zoomable thumbnail image of the rendered page (Figures 1-2), an approach also used in a few other research systems (Hightower et al. 1998; Ayers and Stasko 1995). It also detects the title and URL of the page, and these are (optionally) displayed alongside the thumbnail. Because some thumbnails may be difficult to distinguish from others (such as a site’s pages that follow a standard look, we also provide larger views: mousing over any miniaturised thumbnail causes it to zoom to approximately four times the size (Figure 2a bottom).” [Cockburn, Section 3.1]

Figure 1 below, reproduced from Cockburn, illustrates a thumbnail of the home page of the Department of Computer Science, University of Calgary.



Figure 1: Dog-eared thumbnail.

Webview distinguishes between thumbnails of interior pages vs. home page thumbnails associated with a domain:

To avoid excessive nesting depth in the display, **each new web site that the user visits is added to the display at the top-level of the nesting structure. The figure therefore shows that the “COSC Home Page”, “University of Canterbury...” and “Thimbleby’s home page” are each located at separate web-sites.** Naturally, separating sites in the display will also separate pages on related topics that are stored at different sites. To ease this problem, WebView adds a feature that displays cross-site navigational links. The arrowhead links connecting thumbnails in Figure 3 shows that the user navigated from the “Academic Visitors” page to “Thimbleby’s home page”, and then to the page “Professor Ian. H. Witten...”, each of which are

Cockburn: Webview

stored at different sites.” [Cockburn, Section 3.3, emphasis added]

Figure 2a, which illustrates the above special treatment of home page thumbnails by their placement at the root of each sub-tree, is reproduced below.

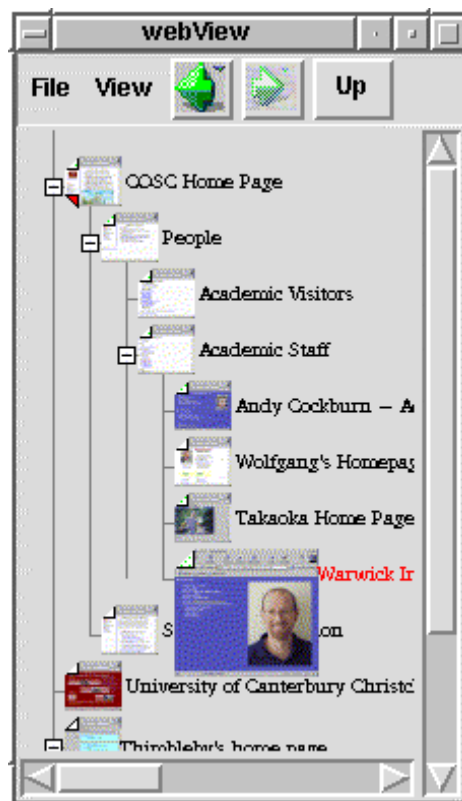


Figure 2a) hub and spoke

Thumbnail visual image of a home page, where the image is of a different page than that referenced by the hyperlink:

Webview has several ways of associating a hyperlink with a thumbnail visual image.

Cockburn: Webview

		<p>First, in the hub-and-spoke view, WebView presents a web pages as a collapsible structural tree (a hierarchy) of visual thumbnails under a common root in the domain path. That is, all pages under a node could be collapsed and represented by that single entry and thumbnail image:</p> <p>“WebView supports two primary display organisation schemes, controlled by options under its ‘View’ menu. Its ‘hub-and-spoke’ view (Figure 2a), displays the tree-like nesting relationship between the storage location of pages. In Figure 2a, for example, it is clear that the user has visited four pages under the “Academic Staff” page. Organising the display as a structural tree, such as this, is intended to help the user see the context of their navigational acts: pages on a similar topic are likely to be displayed in close proximity to each other.” [Cockburn, Section 3.3]</p> <p>In addition, Figure 2a, <i>supra</i>, illustrates this hub-and-spoke view. Clicking the ‘-’ sign left of the thumbnail (such as the COSC Home Page at the top) would collapse all children pages under the COSC Home Page thumbnail image.</p> <p>Second, Cockburn associates multiple links with a single thumbnail, where those links are all page links on the actual page represented by the thumbnail. When a user selected that thumbnail with a right-mouse button press, the popup menu with all alternate links are displayed; the user can then select any link to directly navigate to that page without having to visit the thumbnail’s page:</p> <p>“Finally, WebView parses the page’s HTML contents, extracts the page links, and uses this information to construct a pop-up ‘shortcut’ menu associated with the thumbnail (Figure 4). Clicking on a thumbnail pops up the shortcut menu that can be used to navigate directly to any subordinate child page. The shortcut menu works within both temporal and hub-andspoke views.” [Cockburn, Section 3.3]</p> <p>Figure 2b from [Cockburn] is reproduced below. In this case, the popup menu raised by selecting the thumbnail shows all links associated with the Curriculum Vitae home page. Selecting a link would go directly to that page, even though the thumbnail image was to the Curriculum Vitae home page.</p>
--	--	---

Cockburn: Webview

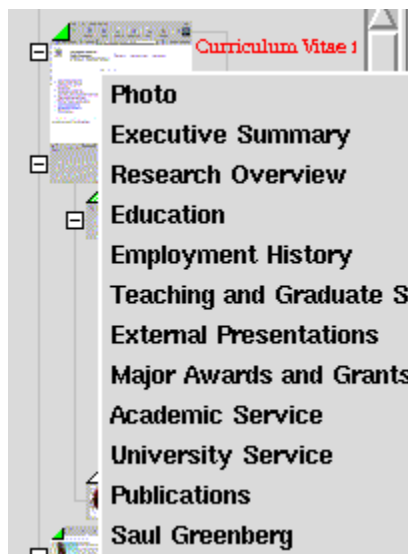


Figure 2b) shortcut pop-up menu

In addition, Cockburn explains how the shortcut pop-up menu worked in practice:

“Users readily adapted to most of WebView’s navigational facilities. For instance, in Task 1b, when navigating to “Wal’s” page from “Andy’s” page, all users immediately selected the shortcut menu off the “Academic Staff” thumbnail without returning to the “Academic Staff” page.” [Cockburn, Section 4.1]

Third, Webview adds arrowhead links to a thumbnail to connect it to other thumbnails that the user had navigated to from that page:

“WebView adds a feature that displays cross-site navigational links. The arrowhead links connecting thumbnails in Figure 3 shows that the user navigated from the Academic Visitors” page to “Thimbleby’s home page”, and then to the page “Professor Ian. H. Witten...”, each of which are stored at different sites. [Cockburn, Section 3.3]

Cockburn: Webview

Figure 3, reproduced below from [Cockburn] illustrates this connection. Note that one link is from 'Thimbleby's home page' to another different 'Professor Ian H. Witten at Wa' home page.

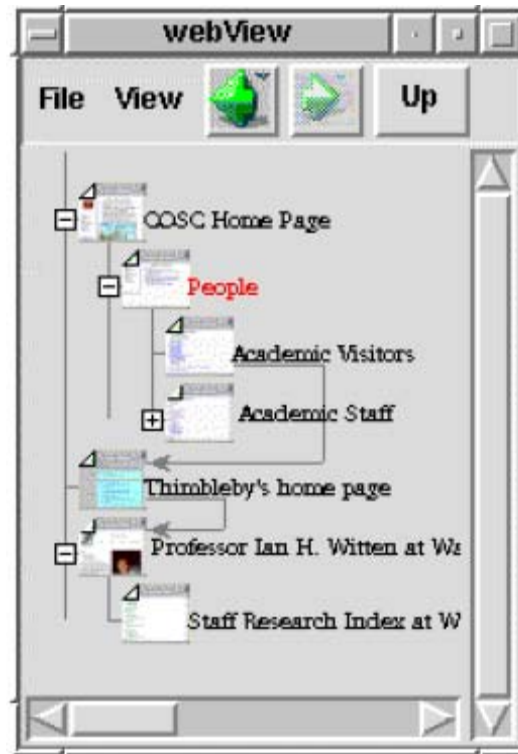


Figure 3: Cross-site link indicators

18. A system for presenting Internet information to a user comprising:

See Claim 1 analysis, *supra*, which is hereby incorporated by reference.

Cockburn: Webview


	first functionality providing to a user a visual image of a web page containing at least one hyperlink; and	
	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image.	

EXHIBIT G6

CNN Interactive: Site-Seer

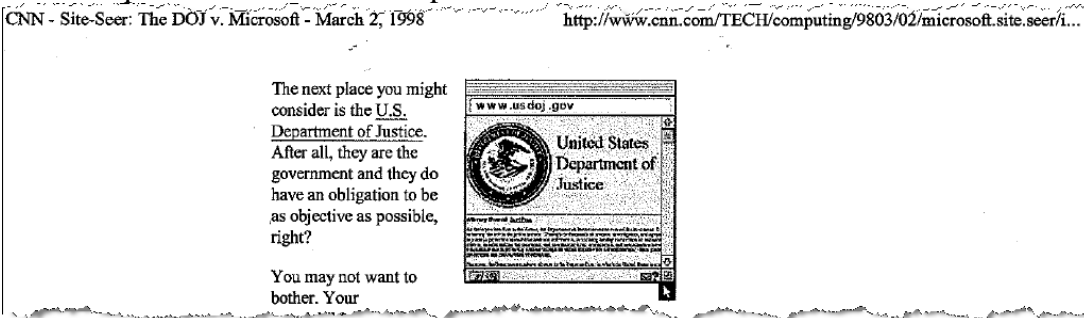
The following claims are invalid as being obvious under CNN Interactive in view of Brown, or Kraft, or Schmid, or Kopetzky

CNN Interactive: Site-Seer: the DOJ v. Microsoft. Dated March 2, 1998 ('CNN Interactive')

Claim #	The '904 Patent	Disclosure of Each Limitation in CNN Interactive ¹
1.	A method for presenting Internet information to a user comprising:	
	providing to a user a visual image of a web page containing at least one hyperlink;	<p><i>CNN Interactive discloses a visual image of a web page containing a hyperlink, a portion which is reproduced below:</i></p>  <p><i>Figure 1.</i></p>

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

CNN Interactive: Site-Seer

	<p>and at least partially concurrently providing a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image.</p>	<p><u>Thumbnail visual image of a home page.</u></p> <p>CNN Interactive discloses including thumbnail visual images of one or more home pages. The thumbnail visual image in Figure 1 Supra represents the home page www.microsoft.com. A fragment of a different part of this page showing this second thumbnail image of a home page representing the home page of the United States Department of Justice is reproduced below.</p> <p>CNN - Site-Seer: The DOJ v. Microsoft - March 2, 1998 http://www.cnn.com/TECH/computing/9803/02/microsoft.site.seer/i...</p>  <p><u>CNN Interactive discloses that hyperlinks on or nearby the thumbnail visual image of a home page may be to the home page itself, or to a different web page than represented by the thumbnail.</u></p> <p>I have examined the HTML surrounding the Microsoft Home Page thumbnail in Figure 1, <i>supra</i>, which reads:</p> <pre><IMG SRC="microsoft.com.jpg" ALT="http://www.microsoft.com/" ALIGN=LEFT WIDTH="199" HEIGHT="208" BORDER="0" HSPACE="10" VSPACE="5"> <P>One of the first sites you might think of visiting is Microsoft. Don't assume you'll be getting an unbiased view in the documents presented in the corporate information area. But you will get some insight</pre>

CNN Interactive: Site-Seer

		<p>into how Microsoft views itself and how it intends to defend itself against the Justice Department's charges.</p></p> <p>This says that the thumbnail image of the Microsoft home page contains a link to the front page of the Microsoft web site: http://www.microsoft.com/. If an end user clicked the thumbnail, they would be directed to the Microsoft home page as specified by the microsoft domain name. This HTML also says that:</p> <ul style="list-style-type: none"> • the thumbnail image is stored in the file “microsoft.com.jpg”, • the text http://www.microsoft.com/ should be displayed if the image is not available, • the ‘Microsoft’ word in the text accompanying and annotating this thumbnail also links to Microsoft front page. • the phrase ‘documents presented in the corporate information area’ in the text accompanying and annotating this thumbnail links to an interior Microsoft page http://www.microsoft.com/corpinfo/ rather than the front page. <p><u><i>CNN discloses employing an image server that stores and provides said thumbnail visual image.</i></u></p> <p>Because the HTML uses the IMG tag to retrieve the Microsoft home page thumbnail image file, one skilled in the art would know that this thumbnail image could reside on any server on the Internet that implements the HTTP protocol; that server would then be considered a separate image server from the server serving this particular CNN web page. The code structured around the US Department of Justice home page thumbnail works in a similar fashion.</p>
18.	A system for presenting Internet information to a user comprising:	See Claim 1 analysis, <i>supra</i> , which is hereby incorporated by reference.
	first functionality providing to a user a visual image of a web page containing at least one hyperlink;	

CNN Interactive: Site-Seer

	and	
	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image.	

EXHIBIT G7

Exhibit 7 - Brown (U.S. Patent No. 6,356,908)

<u>The following claims are invalid as being obvious under Brown in view of Kraft, or Schmid, or Kopetzky</u>		
US Patent No. 6,356,908: <i>Automatic web page thumbnail generation</i> . (Michael Wayne Brown, Kelvin Roderick Lawrence, Michael A. Paolini). Filed Jul. 30, 1999, Issued March 12, 2002.		
Claim #	The '904 Patent	Disclosure of Each Limitation in Brown¹²
1.	A method for presenting Internet information to a user comprising:	“The invention relates generally to the field of computer software and, more specifically, to Internet related computer software. . . . it would be beneficial for Internet users to have a tool to enable them to make more informed decisions about which links to follow.” [Brown at col. 1:23-25; 2:8-10]
	providing to a user a visual image of a web page containing at least one hyperlink;	“FIG. 8 shows a screen image for search results with thumbnails placed in-line near a respective link.” [Brown at col. 2:48-49] See also Figure 8, <i>infra</i> .
	and at least partially concurrently providing a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and	<i>at least partially concurrently providing a thumbnail of the home page of at least one web site which is represented by said at least one hyperlink via the Internet:</i> “In one preferred embodiment, the thumbnails are displayed in-line (that is each thumbnail is placed below the preceding thumbnail in a vertical line) near the corresponding link on the currently displayed web page as illustrated in FIG. 8.”

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

² The '904 Patent cites two patents by Brown: 6,356,908 (the “'908 Patent”) and 6,665,838 (the “'838 Patent”). Only the '838 Patent is discussed in the '904 file history. The '908 Patent, charted here, was “made of record and not relied upon” but was “considered pertinent to applicant’s disclosure.” '904 Patent file history, Office Action dated March 8, 2004, page 11.

Exhibit 7 - Brown (U.S. Patent No. 6,356,908)

	provides said thumbnail visual image.	<p>[Brown at col. 6:33-37]</p> <p><u>Thumbnail visual image of a home page.</u> Brown teaches displaying thumbnails of web pages, which would naturally include home pages as a subset.</p> <p><u>Thumbnail visual image of a home page represented by a hyperlink, but is of a different web page than referenced by the hyperlink.</u> Brown also teaches displaying the home page of at least one web site which is represented by said at least one hyperlink. See Brown at column 8, lines 16-27: “In another embodiment of the pop-up thumbnails, <u>rather than generating and displaying thumbnails of the web pages associated with links, an icon representing the domain of that link could be generated and displayed next to the text representing the link.</u> For example, if the domain is associated with Yahoo, then an icon displaying the Yahoo logo might be displayed next to the link. Furthermore, the icon could be assigned by the user or by the domain itself, and picked up automatically by the browser. This icon would then pop-up next to the link as the pointer moves over the link, just as the thumbnail does in an example depicted in FIG. 12.” (emphasis added).</p> <p>One of ordinary skill in the art would have understood that a thumbnail preview image of a top level domain could serve as an iconic representation of that domain. This is consistent with the use of the term in the literature (e.g., Mullet and Sano’s book ‘Designing Visual Interfaces’ by Mullet and Sano, and US Patent 5,548,692 by Cok et. al.).</p>
18.	A system for presenting Internet information to a user comprising:	See Claim 1 analysis, <i>supra</i> , which is hereby incorporated by reference.
	first functionality providing to a user a visual image of a web page containing at least one hyperlink; and	

Exhibit 7 - Brown (U.S. Patent No. 6,356,908)

	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of the home page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image.	
	said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.	

EXHIBIT G8

Berners-Lee et. al. Hypertext Markup Language

<u>The following claims are invalid as being obvious under Berners-Lee in view of Brown, or Kraft, or Schmid, or Kopetzky</u>		
T. Berners-Lee and D. Connolly. (1995) <i>Hypertext markup Language - 2.0</i> . Network Working Group, Request for Comments: 1866, Category: Standards Track. MIT/W3C. November. ('Berners-Lee')		
Claim #	The '904 Patent	Disclosure of Each Limitation in Berners-Lee¹
35.	A method for presenting Internet information to a user comprising:	<p>Berners-Lee discloses a mechanism and standard for providing images using hyperlinks via the Internet. This standard is the HTML 2.0 standard and is served by the HTTP Protocol. From this specification, any server that responds to the IMG request by providing an image is, by definition, an image server. If that server is located on a machine different than the one serving the web page, then it is, by definition, a separate image server.</p> <p>Section 5.10, which describes the IMG tag, is reproduced below. Note that:</p> <ol style="list-style-type: none"> 1. The SRC of an image as specified by the URI of the image resource may be located on any server. 2. The link associated with the image, when an tag is hyperlinked, as in: ", could be any hyperlink. <p>"5.10. Image: IMG</p> <p>The element refers to an image or icon via a hyperlink (see 7.3, "Simultaneous Presentation of Image Resources").</p>
	providing to a user a visual image of a web page containing at least one hyperlink;	
	and at least partially concurrently providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image,	
	said providing a thumbnail visual image comprising employing a web browser which interfaces via	

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

Berners-Lee et. al. Hypertext Markup Language

	<p>the Internet with a web server, separated from said image server, including visualization functionality, said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.</p>	<p>HTML user agents may process the value of the ALT attribute as an alternative to processing the image resource indicated by the SRC attribute.</p> <p>NOTE - Some HTML user agents can process graphics linked via anchors, but not graphics. If a graphic is essential, it should be referenced from an <A> element rather than an element. If the graphic is not essential, then the element is appropriate.</p> <p>Attributes of the element: SRC specifies the URI of the image resource.</p> <p>ALIGN alignment of the image with respect to the text baseline. * `TOP' specifies that the top of the image aligns with the tallest item on the line containing the image. * `MIDDLE' specifies that the center of the image aligns with the baseline of the line containing the image. * `BOTTOM' specifies that the bottom of the image aligns with the baseline of the line containing the image.</p> <p>ALT text to use in place of the referenced image resource, for example due to processing constraints or user preference.</p> <p>ISMAP indicates an image map (see 7.6, "Image Maps").</p> <p>SRC specifies the URI of the image resource.</p> <p>NOTE - In practice, the media types of image resources are limited to a few raster graphic formats: typically `image/gif', `image/jpeg'. In particular, `text/html' resources are not intended to be used as image resources.</p>
--	---	--

Berners-Lee et. al. Hypertext Markup Language

		<p>Examples of use:</p> <p> Be sure to read these instructions.</p> <p> ” [Berners-Lee, Section 7.3]</p> <p><i>Section 7.3 of the standard also anticipates ‘a multiplicity of downloaders’, where multiple image requests could be sent to one or more servers at the same time. This mean they are handled partially concurrently rather than sequentially.</i></p> <p>“7.3. Simultaneous Presentation of Image Resources An HTML user agent may activate hyperlinks indicated by and <INPUT> elements concurrently with processing the document; that is, image hyperlinks may be processed without explicit request by the user. Image resources should be embedded in the presentation at the point of the tail anchor, that is the or <INPUT> element.” [Berners-Lee, Section 7.3]</p>
46.	A system for presenting Internet information to a user comprising:	<i>See Claim 35 supra, which is hereby incorporated by reference.</i>
	first functionality providing to a user a visual image of a web page containing at least one hyperlink; and	
	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one hyperlink via the Internet by	

Berners-Lee et. al. Hypertext Markup Language

	employing an image server that stores and provides said thumbnail visual image, said second functionality comprising third functionality employing a web browser which interfaces via the Internet with a web server, separated from said image server, including visualization functionality,	
	said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.	

EXHIBIT G9

Hess (U.S. Patent No. 6,058,417)

<u>The following claims are invalid as being obvious under Hess in view of Kraft or Kopetzky</u>		
US Patent 6,058,417: <i>Information Presentation and Management in an Online Trading Environment</i>. (Martin Hess, Michael Wilson) Filed Oct. 23, 1998, Issued May 2, 2000. ('Hess')		
Claim #	The '904 Patent	Disclosure of Each Limitation in Hess¹
16.	A method for generating an image server database of thumbnail visual images of web pages, the method comprising:	<p>Hess discloses a method for generating an image server database of thumbnail visual images:</p> <p>“A method and apparatus for information presentation and management in an online trading environment are described. According to one aspect of the present invention, person-to-person commerce over the Internet is facilitated by providing prospective buyers the ability to quickly preview items for sale. Images are harvested from a plurality of sites based on user-supplied information. The user-supplied information includes descriptions of items for sale and locations from which images that are to be associated with the items can be retrieved. Thumbnail images are created corresponding to the harvested images and are aggregated onto a web page for presentation at a remote site.” [Hess at col. 2:10-21]</p> <p>“Importantly, as one feature of the present embodiment, thumbnail images are not stored as individual files; rather, they are stored in an efficient database format that will be described further below.” [Hess at col. 5:42-45]</p> <p>Although Hess is not specifically directed towards thumbnail images of web pages, it discloses such an application:</p>

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

Hess (U.S. Patent No. 6,058,417)

		<p>“Importantly, while embodiments of the present invention will be described with respect to an online person-to-person trading environment, the method and apparatus described herein are equally relevant to other applications in which image data is collected from disparate sources and presented to a user . . .” [Hess at col. 3:65-4:4]</p> <p>Additionally, one of ordinary skill in the art would have known that a web page could consist of a single image file, i.e. “image.jpg,” linked to directly.</p>
	receiving a list of URLs corresponding to said web pages, the thumbnail visual images of which it is desired to supply to said image server database;	<p>Hess discloses receiving a list of URLs to be supplied as thumbnail images to an image server database:</p> <p>“FIG. 7 is a flow diagram illustrating the image harvesting process according to one embodiment of the present invention. At step 710, image location information is retrieved from the listing database 420 for a set of images that will be downloaded concurrently. According to one embodiment, the image location information is a URL.” [Hess at col. 7:42-48]</p> <p>See also, Figure 7:</p>

Hess (U.S. Patent No. 6,058,417)

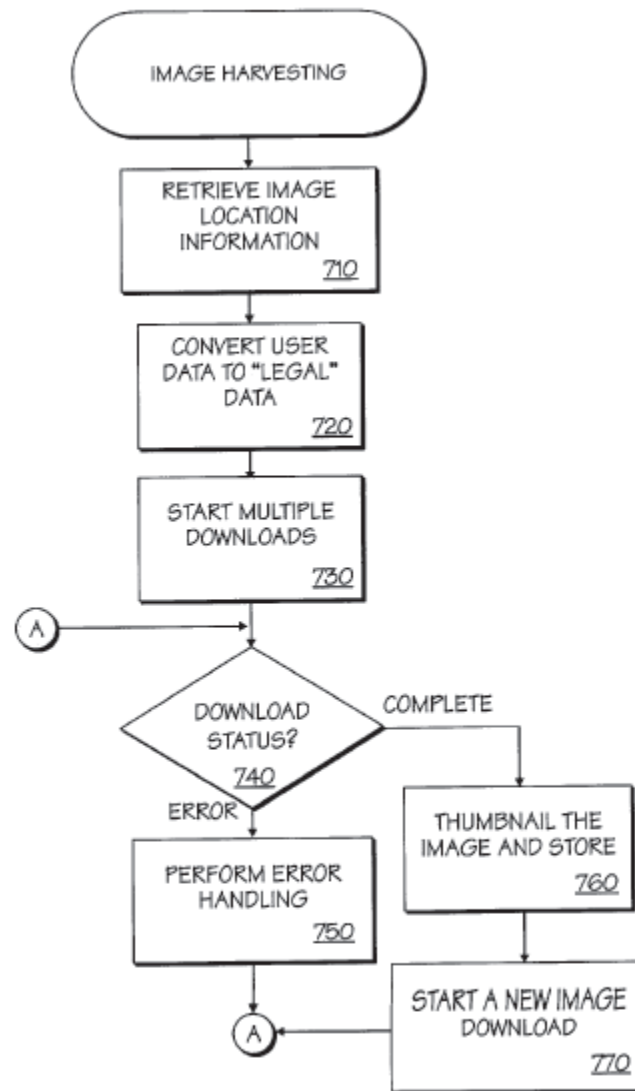


Fig. 7

Hess (U.S. Patent No. 6,058,417)

	operating a multiplicity of downloaders simultaneously to retrieve from the Internet, web pages and embedded objects corresponding to URLs from said list;	<p>Hess discloses operating a multiplicity of downloaders simultaneously to retrieve from the Internet the objects in the provided URL list:</p> <p>“At step 730, multiple image downloads are started using a sockets-based interface. Prior to starting the downloads, it may be necessary to attempt a variety of option configurations in order to establish communication with a particular server. At any rate, assuming communication has successfully been established with the servers that have the desired image data, in one embodiment, 500 downloads are performed concurrently.” [Hess at col. 7:66-8:7]</p> <p>See also [Hess, Fig. 7], <i>supra</i>, box 730 labeled ‘Start multiple downloads’</p>
	causing a thumbnail generator to render retrieved web pages retrieved simultaneously by said multiplicity of downloaders; and	<p>Hess discloses rendering retrieved web page images</p> <p>“As will described further below, the harvesting process 455 automatically downloads the specified image, <u>converts it to the appropriate format</u>, and scales it to the appropriate size that is appropriate for use with the Gallery presentation mechanism.” [Hess at col. 7:10-14] (emphasis added)</p> <p>See also [Hess, Fig. 7], <i>supra</i>, box 730 labeled ‘Start multiple downloads’, which leads into box 740 labeled ‘Download Status’ which leads (when a download is complete) to Box 760 labeled ‘Thumbnail The Image and Store’</p>

Hess (U.S. Patent No. 6,058,417)

	<p>causing said thumbnail generator to shrink said rendered images of said retrieved web pages and supply them to said image server database.</p>	<p>Hess discloses a thumbnail generator that shrinks the rendered images and supplies them to an image server database:</p> <p>“The thumb building machine 450 includes a harvesting process 455 and a database 460. As will be described further below, the harvesting process 455 periodically harvests images that sellers have associated with items in the listing database 420. After a set of images have been harvested and thumbnailed, the harvesting process 455 notifies the thumb server 430 that new thumbnails are available.” [Hess at col. 5:25-31]</p> <p>See Figure 4:</p>
--	---	---

Hess (U.S. Patent No. 6,058,417)

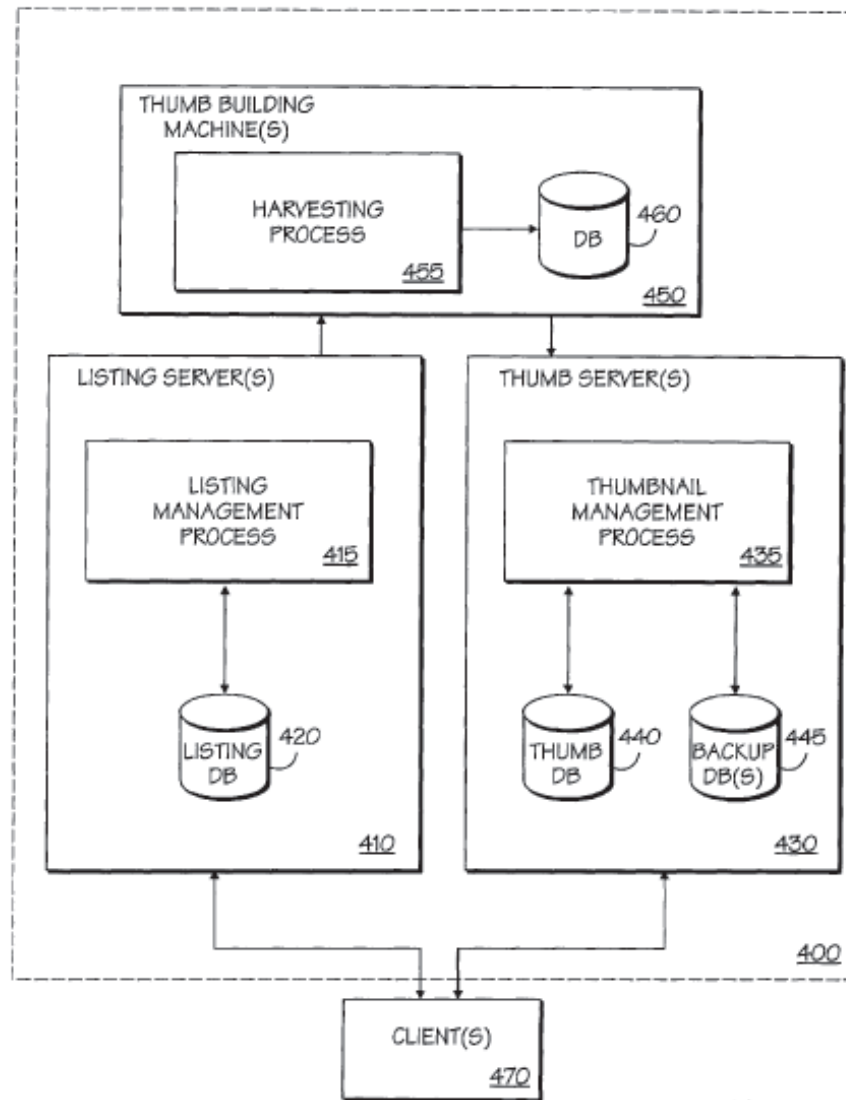
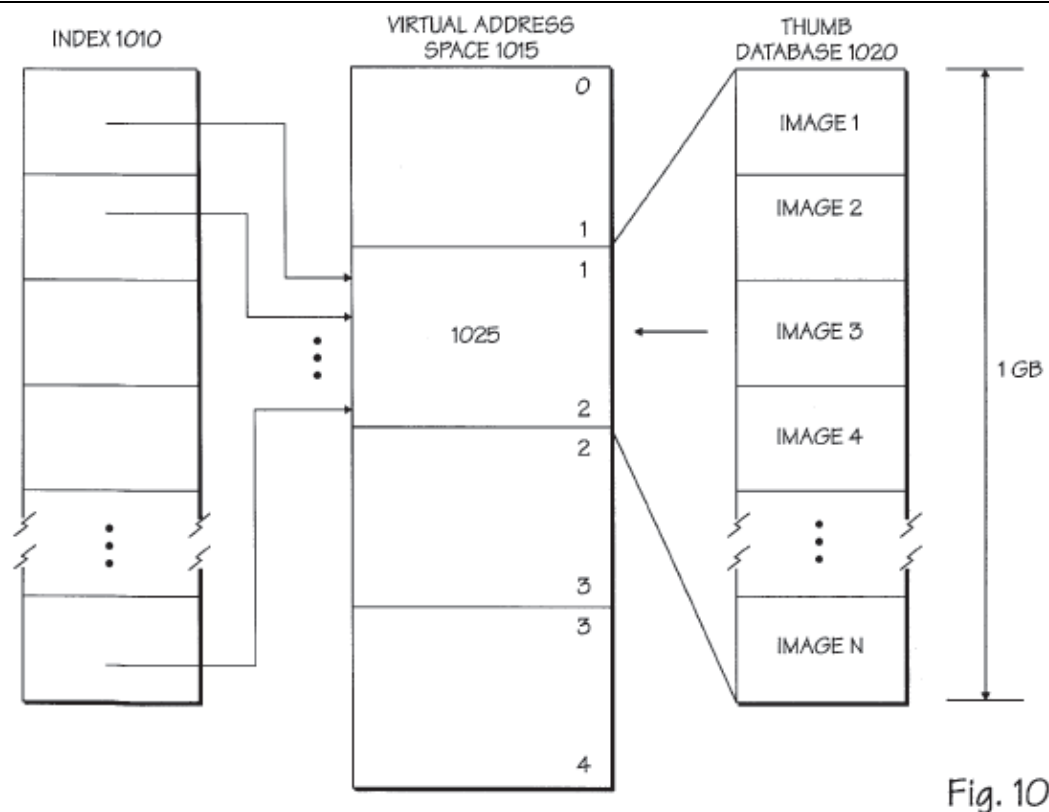


Fig. 4

Hess (U.S. Patent No. 6,058,417)

		<p>Additionally:</p> <p>“At step 760, the one or more images that have been downloaded successfully are thumbnailled and stored for later inclusion in the thumbnail database 440. . . .</p> <p>Thumbnailing an image may be broken down into three steps: (1) <u>first, decompression is performed from the harvested image’s source format</u>; (2) <u>then, the decompressed image is converted to a thumbnail</u> that will fit within a predetermined space. For example, the largest dimension of the source image may be scaled to fit the corresponding dimension of the predetermined space, then the other dimension of the source image may be scaled proportionately; (3) finally, the thumbnail is recompressed into a predetermined output format, e.g., Joint Photographics Expert Group (JPEG).” [Hess at col. 8:20-35] (emphasis added)</p> <p>The image server database is described as follows:</p> <p>“FIG. 10 illustrates memory mapped file access to the thumb database 1020 according to one embodiment of the present invention. According to the embodiment depicted, rather than maintaining a complex file structure with potentially millions of separate files, a simple and efficient thumb database 1020 is maintained that is designed to get information into memory quickly to provide fast access to the thumbnail images stored therein. Briefly, the goal is to keep the thumb database 1020 reasonably sized so that it can be completely loaded into a virtual address space 1015 and accessed as a memory mapped file.” [Hess at col. 10:6-16]</p> <p>See also, Figure 10:</p>
--	--	---

Hess (U.S. Patent No. 6,058,417)



See also [Hess, Fig. 7], *supra*, box 730 labeled 'Start multiple downloads', which leads into box 740 labeled 'Download Status' which leads (when a download is complete) to Box 760 labeled 'Thumbnail The Image and Store'

17.	A method according to claim 16 also comprising deleting executable content from said retrieved web pages.	The thumbnail generated image of the retrieved object necessarily deletes executable content, as images do not contain executable content.
-----	--	--

Hess (U.S. Patent No. 6,058,417)

33.	A system for generating an image server database of thumbnail visual images of web pages, the system comprising:	See Claim 16 analysis, <i>supra</i> , which is hereby incorporated by reference
	a multiplicity of downloaders, each receiving at least one URL from a list of URLs corresponding to said web pages, the thumbnail visual images of which it is desired to supply to said image server database, and simultaneously retrieving from the Internet web pages and embedded objects corresponding to said at least one URL; and	
	a multiplicity of downloaders, each receiving at least one URL from a list of URLs corresponding to said web pages, the thumbnail visual images of which it is desired to supply to said image server database, and simultaneously retrieving from the Internet web pages and embedded objects corresponding to said at least one URL; and	
	at least one thumbnail generator operative to render the web pages, shrink said rendered images of the web pages and supply said rendered images to said image	

Hess (U.S. Patent No. 6,058,417)

	server database.	
34.	A system according to claim 33 and wherein said multiplicity of downloaders are operative to delete executable content from the web pages.	See Claim 17 analysis, <i>supra</i> , which is hereby incorporated by reference

EXHIBIT G10

Frankel: WebSeer

**The following claims are invalid as being obvious under Frankel in view of
Kraft or Kopetzky**

Frankel, C., Swain, M. J., and Athitsos, V., *Webseer: an Image Search Engine for the World Wide Web*. Technical Report. UMI Order Number: TR-96-14., University of Chicago (1996). ('Frankel')

Claim #	The '904 Patent	Disclosure of Each Limitation in Frankel ¹
16.	A method for generating an image server database of thumbnail visual images of web pages, the method comprising:	<p><i>Frankel's Webseer system is directed to generating an index (an image server database) of images comprising thumbnails:</i></p> <p>"...while Web pages typically contain both text <i>and</i> images, all the currently available search engines only index text. This paper describes <i>WebSeer</i>, a system for locating images on the Web. WebSeer uses image content in addition to associated text to index images, presenting the user with a selection that potentially fits her needs. [Frankel, Abstract]"</p> <p><i>Frankel discloses that Webseer is directed to generating an image server database of thumbnails:</i></p> <p>"Scaling Up. Techniques for content-based retrieval that work with a hundred, a thousand, or even ten thousand images will not necessarily scale up to the task of indexing all the images on the World Wide Web. There are an estimated 30 million <i>Web pages</i> (HTML documents). Our preliminary experiments indicate there may be about one-third as many images as Web pages, meaning about 10 million images to index. Rough calculations suggest that WebSeer's database will be about 1.5 GB, and storing thumbnails will take up about 15 GB of disk space. Crawling the Web to index all the images will require downloading them all.. ..." [Frankel, p. 18, emphasis added]</p>

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

Frankel: WebSeer

		<p><i>Frankel discloses that Webseer is an image server:</i></p> <p>“The WebSeer Search Server accepts requests from the WebSeer CGI script and performs the appropriate searches based on the form fields which the user has filled in.” [Frankel, p. 7, emphasis added]”</p> <p><i>Frankel discloses that the image server returns a selectable thumbnail image and associated URL which, depending how it is selected, will display in a browser either the full sized image or the web page containing that image:</i></p> <p>“A user is interested in finding small, close-up images of Rebecca De Mornay. They type “Rebecca De Mornay” as the search text, and make selections as displayed in Figure 1. The results page interface, shown in Figure 2, works as follows. Thumbnails of the resulting images are displayed above a bar which indicates the size of the original image. Clicking on the image will launch your browser to the URL original image. Clicking on the page icon to the right of the image will launch your browser to the URL of the page which contains the image. [Frankel, pp. 9-10, emphasis added]</p>
	receiving a list of URLs corresponding to said web pages, the thumbnail visual images of which it is desired to supply to said image server database;	<p><i>Frankel discloses a URL Server receiving a list of URLs to be supplied as thumbnail images to an image server database. Notice that URLs is plural rather than singular.</i></p> <p>“The URL Server is a multi-threaded java application which receives requests to download URLs from the WebSeer Crawler. Separating the URL server from the Crawler application allows us to download pages from multiple machines (with different operating systems) simultaneously.” [Frankel, p. 7, emphasis added]</p>
	operating a multiplicity of downloaders simultaneously to retrieve from the Internet, web pages and embedded objects corresponding to URLs from said	<p><i>Frankel discloses operating a multiplicity of downloaders simultaneously to retrieve web pages and images from web pages from the Internet:</i></p> <p>“Scaling Up Techniques for content-based retrieval that work with a hundred, a thousand, or even</p>

Frankel: WebSeer

list;	<p>ten thousand images will not necessarily scale up to the task of indexing all the images on the World Wide Web. There are an estimated 30 million <i>Web pages</i> (HTML documents). Our preliminary experiments indicate there may be about one-third as many images as Web pages, meaning about 10 million images to index. Rough calculations suggest that WebSeer's database will be about 1.5 GB, and storing thumbnails will take up about 15 GB of disk space. Crawling the Web to index all the images will require downloading them all. Our current multi-threaded Web crawler can download many pages per second, running on a 200 MHz Pentium Pro PC attached to a dual T1 line shared with the rest of the University. ..." [Frankel, p. 18, emphasis added]</p> <p>"With the exception of the URL Server, which is written in Java, all executables are written in C++ and run on a M.S. Windows NT 3.51 platform.</p> <ol style="list-style-type: none"> 1. The WebSeer Crawler crawls the web downloading both HTML pages and images. The crawler is multi-threaded so that the delay downloading pages is spread over multiple threads. Each thread is connected to a database of previously visited (and waiting to be visited) URLs using the ODBC 2.0 database protocol. 2. The URL Server is a multi-threaded java application which receives requests to download URLs from the WebSeer Crawler. Separating the URL server from the Crawler application allows us to download pages from multiple machines (with different operating systems) simultaneously. 3. The WebSeer Indexer creates the index which is searched by the users. The indexer parses the HTML code and executes the appropriate image understanding applications. 4. The WebSeer CGI script is called when the user submits (POSTs) a query from the WebSeer form. This script opens a TCP/IP connection to the WebSeer Search Server, and formats the results for display to the user. 5. The WebSeer Search Server accepts requests from the WebSeer CGI script and performs the appropriate searches based on the form fields which the user has filled in." [Frankel, p. 7, emphasis added] <p>"Since the WebSeer Crawler is designed to download images, we obey the restrictions specified by the <code>robots.txt</code> file when deciding whether to download</p>
-------	---

Frankel: WebSeer

		<p>an html file, but not when downloading an image file...” [Frankel, p. 7]</p> <p><i>Webseer’s crawler is also multi-threaded and thus operates a multiplicity of downloaders simultaneously. It also stores thumbnails in this database:</i></p> <p>Scaling Up</p> <p>Techniques for content-based retrieval that work with a hundred, a thousand, or even ten thousand images will not necessarily scale up to the task of indexing all the images on the World Wide Web. There are an estimated 30 million <i>Web pages</i> (HTML documents). Our preliminary experiments indicate there may be about one-third as many images as Web pages, meaning about 10 million images to index. Rough calculations suggest that WebSeer’s database will be about 1.5 GB, and storing thumbnails will take up about 15 GB of disk space. Crawling the Web to index all the images will require downloading them all. Our current multi-threaded Web crawler can download many pages per second, running on a 200 MHz Pentium Pro PC attached to a dual T1 line shared with the rest of the University. [Frankel, p. 18]</p> <p><i>Frankel discusses the reasons behind using a multiplicity of downloaders:</i></p> <p>“TheWebSeer Crawler is separated into a number of different executables with efficiency in mind. The crawler is a multi-threaded Windows NT executable which establishes connections to the database and the image processing server during startup. The crawler is designed so that multiple copies of the crawler can be run on different PCs simultaneously. The amount of required communication between threads on a single crawler and between crawlers on different machines is kept to minimum. The advantage is that while some threads are busy waiting to download images or waiting for the image processing server to return results, other threads can be parsing html files or creating thumbnails of the images.</p> <p>Each thread on each crawler acts largely as an independent entity. If a crawler thread is connected to an image processing server thread, and that image processing server machine crashes (e.g. runs out of memory), then only that crawler thread will be in jeopardy. Crawler threads connected to image processing servers on different machines will not be effected. Additionally, if a crawler thread notices that the connection to it’s image processing server is broken, the crawler thread will first attempt to reconnect to the same image processing server and, if that fails, it will</p>
--	--	---

Frankel: WebSeer

		attempt a connection to a different image processing server.” [Frankel, Section 6.2]
	causing a thumbnail generator to render retrieved web pages retrieved simultaneously by said multiplicity of downloaders; and	<p><i>Frankel discloses rendering retrieved images from web pages as thumbnails:</i></p> <p>“TheWebSeer Crawler is separated into a number of different executables with efficiency in mind. The crawler is a multi-threaded Windows NT executable which establishes connections to the database and the image processing server during startup. The crawler is designed so that multiple copies of the crawler can be run on different PCs simultaneously. The amount of required communication between threads on a single crawler and between crawlers on different machines is kept to minimum. The advantage is that while some threads are busy waiting to download images or waiting for the image processing server to return results, other threads can be parsing html files or creating thumbnails of the images.” [Frankel, Section 6.2, emphasis added]</p> <p>“Rough calculations suggest that WebSeer’s database will be about 1.5 GB, and storing thumbnails will take up about 15 GB of disk space. Crawling the Web to index all the images will require downloading them all. Our current multi-threaded Web crawler can download many pages per second, running on a 200 MHz Pentium Pro PC attached to a dual T1 line shared with the rest of the University. ...” [Frankel, p. 18]</p>

Frankel: WebSeer

	causing said thumbnail generator to shrink said rendered images of said retrieved web pages and supply them to said image server database.	<p><i>Frankel discloses that Webseer's database stores thumbnail images</i> “Rough calculations suggest that to index the entire Web, WebSeer's database would be about 5 GB, and storing thumbnails would take up about 45 GB of disk space.” [Frankel, Section 7]</p> <p><i>Frankel discloses that Webseer database is currently a commercial database engine</i> “Since indexing occurs off-line, the performance of the indexing engine is less crucial than that of the search engine. ... Second, we tried to incorporate standard commercial software and hardware whenever possible. Much work has been put into developing advanced database engines, and WebSeer's ability to leverage technology adds significantly to its power.” [Frankel, p. 6, emphasis added]</p> <p>“The commercial database has been the bottleneck up to this point, though as we optimize its operation, transfer some of the data to custom data structures, and increase the capabilities of its host, we may find that connectivity to the Internet is the bottleneck.” [Frankel, Section 7]</p>
17.	A method according to claim 16 also comprising deleting executable content from said retrieved web pages.	<p><i>Because Webseer parse pages to extract images, by definition it deletes the executable content</i> “3) The WebSeer Indexer creates the index which is searched by the users. The indexer parses the HTML code and executes the appropriate image understanding applications.” [Frankel, p. 7, emphasis added]</p>
33.	A system for generating an image server database of thumbnail visual images of web pages, the system comprising:	See Claim 16 analysis, <i>supra</i> , which is hereby incorporated by reference
	a multiplicity of downloaders, each receiving at least one URL from a list of URLs corresponding to said web pages, the thumbnail	

Frankel: WebSeer

	visual images of which it is desired to supply to said image server database, and simultaneously retrieving from the Internet web pages and embedded objects corresponding to said at least one URL; and	
	a multiplicity of downloaders, each receiving at least one URL from a list of URLs corresponding to said web pages, the thumbnail visual images of which it is desired to supply to said image server database, and simultaneously retrieving from the Internet web pages and embedded objects corresponding to said at least one URL; and	
	at least one thumbnail generator operative to render the web pages, shrink said rendered images of the web pages and supply said rendered images to said image server database.	
34.	A system according to claim 33 and wherein said multiplicity of downloaders are operative to delete executable content from the web pages.	See Claim 17 analysis, <i>supra</i> , which is hereby incorporated by reference

EXHIBIT G11

Akamai's Freeflow

**The following claims are invalid as being obvious under Akamai in view of
Brown, or Kraft, or Schmid, or Kopetzky**

The materials below both describe the Akamai FreeFlow system.

1. Akamai Technologies, Inc. Akamai FreeFlow brochure. Copyright 1999. ('Akamai')
2. Cisco Systems, Inc. The Network Architecture Behind NetAid. White Paper, Copyright 1999, 17 pages. ('Cisco')

Claim #	The '904 Patent	Disclosure of Each Limitation in Akamai ¹
35.	<p>A method for presenting Internet information to a user comprising:</p> <p>providing to a user a visual image of a web page containing at least one hyperlink;</p> <p>and at least partially concurrently providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail</p>	<p>Akamai, Inc. from Cambridge, Mass. business is based around image servers that are separate from the web site and web server that delivers the web page. In particular, they describe FreeFlow as a commercial content delivery system that delivers rich media content, including graphics. As described in their 1999 brochure, 3rd party companies – Akamai clients – can:</p> <p>“... migrate all or selected portions of content to be served by the Akamai network with an easy-to-use software utility called FreeFlow Launcher. FreeFlow Launcher tags objects within a Web page that are to be served over the FreeFlow network. When customers request those objects, the Akamai network serves them from the closest-available, highest-performing server rather than from some distant or overloaded server.”[Akamai, p. 1]</p> <p>Using Akamia FreeFlow, the web site owner still serves the original web pages,</p>

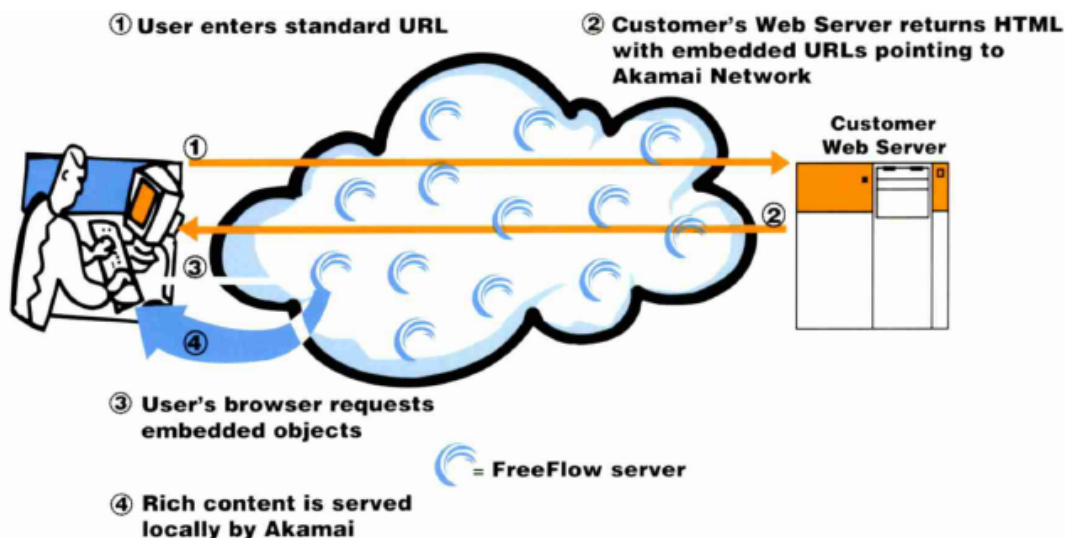
¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

Akamai's Freeflow

visual image,

said providing a thumbnail visual image comprising employing a web browser which interfaces via the Internet with a web server, separated from said image server, including visualization functionality, said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.

while objects within the page – rich media such as images, video and animation – are delivered by “a distributed global network of servers that work together to place content close to Web users.” [Akamai]. Thus FreeFlow’s architecture and indeed its business model is based on having image servers separated from the web servers delivering the actual web page. Figure 3 in Akamai-1 is reproduced below, where it characterizes Internet content delivery with Freeflow. Steps 1 and 2 shows how the user enters the standard URL, and how the customer’s web server returns the HTML with embedded URLs pointing to the Akamai Network. The user’s browser then requests these embedded objects from the Akamai Network, which returns the rich content by a local Akamai server.



Cisco Systems, Inc. further elaborated on the capabilities of Akamai's architecture, and how it proved effective in delivering multimedia content to a very large audience. Cisco, along with the United Nations Development Program, was developing the NetAid web site, where its primary goal was “to raise awareness and foster enhanced communications over the Internet to contribute to the fight against poverty” [Cisco, p. 1].

Akamai's Freeflow

		<p>Part of this website included three concerts broadcast over the Internet on October 9, 1999. Because this website had to handle large amounts of traffic (more than 40 million hits were recorded during the concert events, with Cisco claiming these being record traffic numbers), Cisco paid special attention to the way information could be effectively and reliably delivered to site visitors. In particular, they used Akamai Technologies servers to manage the site Web graphics:</p> <p>“Akamai Technologies had 1200 servers in place around the world that were available to the NetAid project to accelerate all graphics requests.” [Cisco, p. 3].</p> <p>Their figure on page 4 and its caption explains how Cisco hosted the text portion of the Web site to ensure speed, quality and efficiency, while Akamai Technologies served the images for the Web site content using their distributed network:</p> <p>“Overview of Network(s) for netaid.org Functional Divisions. During the concert, the netaid.org Web site was distributed into four separate networks for hosting text, downloading images, streaming video, and handling the e-commerce contribution collection. This unique network design minimized and quickly disbursed areas of congestion. In addition, Multicast capability was available to Internet2 universities.[Cisco, Figure caption on p. 4]</p> <p>The effect is summarized on [Cisco, p. 7]: “the NetAid Web site offloaded graphics requests to Akamai servers to reduce traffic load on the primary Web servers by as much as 90-percent”.</p> <p>The Cisco Systems white paper elaborates on their motivation as well as how this worked under the covers:</p> <p>The Cisco design team was very concerned about their ability to scale the Web site. Graphics on the site could significantly slow performance. Each Web page at the site had nine to 30 graphics. With thousands of potential visitors downloading the same pages at once, there was a strong potential for delay if all content were stored on a single server.</p>
--	--	---

Akamai's Freeflow

		<p>Fortunately, the Web offers a workaround for performance. When a browser first requests a page, it contains a plaintext HTML file with text and page layout and references to inline images. Each image is individually requested by the browser and seamlessly laid out on the screen as a single, integrated "page." These inline references can point anywhere. Inline graphic references on every HTML page in the NetAid Web site were modified during the content-distribution process using the Akamai Launcher script. For example, an HTML tag that looked like:</p> <pre></pre> <p>became</p> <pre></pre> <p>A browser calling up a page at the NetAid Web site automatically requested graphics that were served from the Akamai distributed network of 1200 FreeFlow servers around the world.... [Cisco, p. 8]</p>
46.	A system for presenting Internet information to a user comprising:	<i>See Claim 35 supra, which is hereby incorporated by reference.</i>
	first functionality providing to a user a visual image of a web page containing at least one hyperlink; and	
	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that	

Akamai's Freeflow

	stores and provides said thumbnail visual image, said second functionality comprising third functionality employing a web browser which interfaces via the Internet with a web server, separated from said image server, including visualization functionality,	
	said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.	

EXHIBIT G12

Sclaroff: ImageRover

**The following claims are invalid as being obvious under Sclaroff in view of
Kraft or Kopetzky**

Sclaroff, S., Taycher, L., and La Cascia, M. *ImageRover: A Content-Based Image Browser for the World Wide Web*. Proc. IEEE Workshop on Content-based Access of Image and Video Libraries, June, 1997. Also as Technical Report BU CS TR97-005, Boston University, 1997. ('Sclaroff')

Claim #	The '904 Patent	Disclosure of Each Limitation in Sclaroff¹
16.	A method for generating an image server database of thumbnail visual images of web pages, the method comprising:	<p>Sclaroff discloses a method for generating an image server database of thumbnail images rendered from images gathered from web pages.</p> <p>Sclaroff discloses the relationship between the image server database of thumbnail visual images and the image server:</p> <p>“The image query subsystem is based on a client-server architecture.... Once initialized, the index server runs as a process separate from the database query server, possibly on a different computer. For each query, a client connects to the server to send the query data and then waits for the resulting <i>k</i> nearest neighbors. The server performs the query and returns the results to the client.” [Sclaroff, p. 5]</p> <p>Sclaroff’s Figure 2 reproduced below illustrates this, where we see clients connecting to one or more query servers, which queries the image thumbnail database server, which returns the thumbnails and associated URLs:</p>

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

Sclaroff: ImageRover

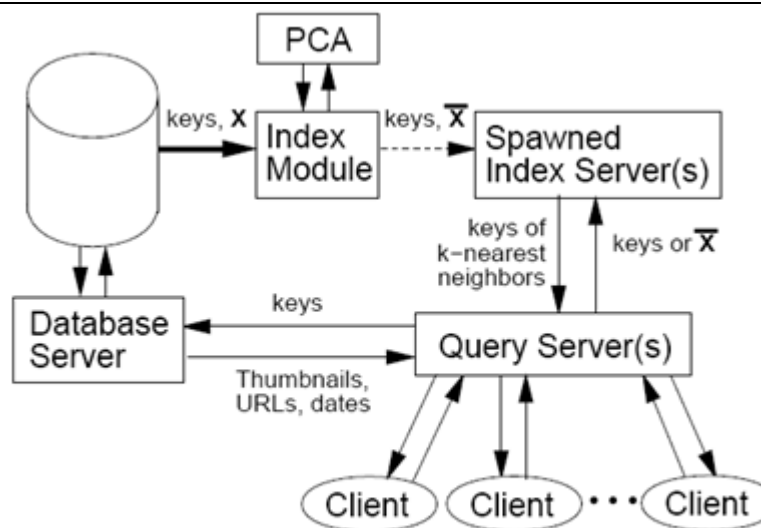


Figure 2: Interactive image query subsystem diagram.

In addition, Sclaroff discloses that image thumbnails are associated with a hyperlink:

“Candidate thumbnail images are returned and displayed as part of an end-user query: ‘each thumbnail image is a hypertext link to the original image. By clicking on any thumbnail, the user can retrieve the desired image from its corresponding home WWW site.’” [Sclaroff, p. 5]

Sclaroff: ImageRover

	<p>receiving a list of URLs corresponding to said web pages, the thumbnail visual images of which it is desired to supply to said image server database;</p>	<p>Sclaroff discloses collection modules that recursively parse and traverse WWW documents, thus receiving the list of URLs corresponding to web pages; note that the production of the thumbnail images are done in the digestion module:</p> <p>“As shown in the Figure 1, robots can contain collection modules, digestion modules, and a local database. The collection modules recursively parse and traverse WWW documents, gathering images. The digestion modules then process these images to extract needed image indexing information and to compute a reduced resolution thumbnail image. The robots are dispatched and coordinated via a separate coordination layer, which also manages updates of the image index database.” [Sclaroff, p. 3, emphasis added]</p> <p>Figure 1 from Sclaroff is reproduced below.</p>
--	--	--

Sclaroff: ImageRover

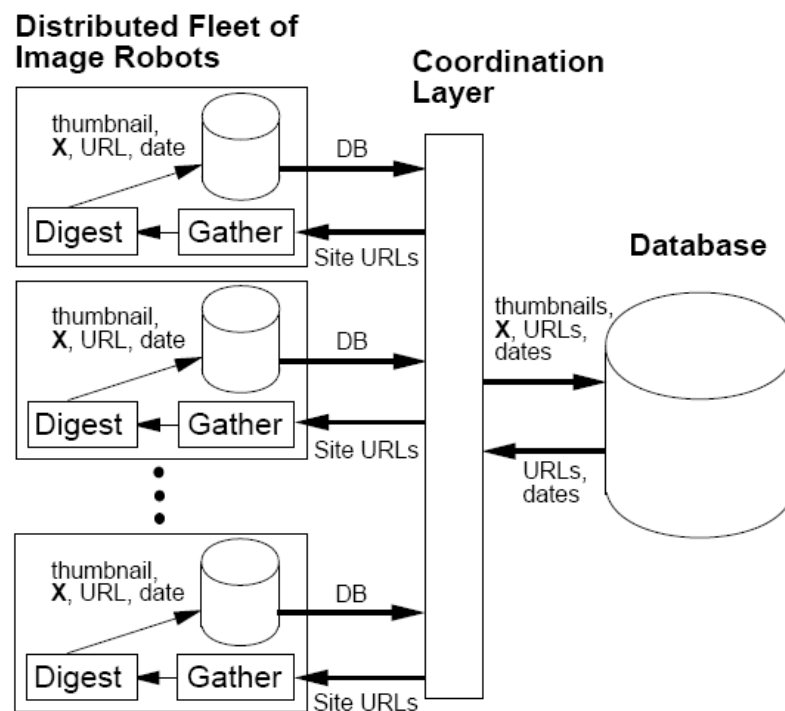


Figure 1: Image robot subsystem diagram. The image collection subsystem utilizes a distributed fleet of WWW robots running on different computers. Robots can contain *collection modules*, *digestion modules*, and a local database. The collection modules recursively parse and traverse WWW documents, gathering images. The digestion modules then process these images to extract needed image indexing information **X** and to compute a reduced resolution thumbnail image. The robots are dispatched and coordinated via a separate *coordination layer*, which also manages updates of the image index database.

Sclaroff discloses supplying images to said image server database:

Sclaroff: ImageRover

		<p>“As shown in the Figure 1, robots can contain collection modules, digestion modules, and a local database. The collection modules recursively parse and traverse WWW documents, gathering images. The digestion modules then process these images to extract needed image indexing information and to compute a reduced resolution thumbnail image. The robots are dispatched and coordinated via a separate coordination layer, which also manages updates of the image index database.” [Sclaroff, p. 3]</p> <p>See Figure 1, <i>supra</i>, its caption, and further descriptions below to see how these thumbnail visual images are supplied to an image server database.</p>
--	--	--

Sclaroff: ImageRover

	<p>operating a multiplicity of downloaders simultaneously to retrieve from the Internet, web pages and embedded objects corresponding to URLs from said list;</p>	<p>Sclaroff discloses operating a multiplicity of downloaders simultaneously to retrieve images from the Internet, as well as the reasons behind using such an approach:</p> <ol style="list-style-type: none"> 1. “Image Collection. In our experiments it has been observed that a single-threaded robot can traverse the web gathering images at an average rate of one image every 82 seconds. Gathering 10-30 million images has the potential to take on the order of 25 years with a single-threaded robot on a single computer. We address this problem by employing a fleet of robots distributed across many machines. Experiments indicate that our framework can allow a modest fleet of 32 robots to collect over one million images monthly. [Sclaroff, p. 2, emphasis added] 2. Image Digestion. As images are gathered, the then needs to digest each image, extract the image statistics and decompositions, and create reduced resolution image thumbnail. While the computation time needed to accomplish this dependends [sic] on the processor and algorithms employed, it is clear a single computer cannot digest 10-30 million in a reasonable amount of time. This hurdle cleared using our multi-processor approach.” [Sclaroff, p. 2, emphasis added] <p>Later, Sclaroff verifies the performance of operating a multiplicity of downloaders, and adds that this could be improved even further if the robots were multi-threaded (another way of implementing a multiplicity of downloaders):</p> <p>“On average, each single-threaded robot can collect 1044 images daily. It is therefore reasonable to expect that a modest fleet of 32 single-threaded robots can collect approximately 1 million images monthly. Multi-threaded robots should achieve significantly greater throughput.” [Sclaroff, p. 7]</p> <p>In addition to the above, Sclaroff describes two methods for implementing a multiplicity of downloaders, i.e., on the same computer or on different computers:</p> <p>“The image collection subsystem utilizes a distributed fleet of WWW robots running on different computers. These robots can be run on a number of computers at a single site (as has been the case in the development of our initial system) or across a</p>
--	---	--

Sclaroff: ImageRover

		<p>number of geographically-distributed computers at volunteer sites.” Sclaroff, p. 3]</p> <p>See Figure 1, <i>supra</i>, which illustrates and describes in its caption how a distributed fleet of WWW image robots running on different computers gather and processes images into thumbnail images and stores them in a database.</p>
	<p>causing a thumbnail generator to render retrieved web pages retrieved simultaneously by said multiplicity of downloaders; and</p>	<p>Sclaroff discloses rendering images from web pages retrieved simultaneously by said multiplicity of downloaders:</p> <p>2. Image Digestion. As images are gathered, the then needs to digest each image, extract the image statistics and decompositions, and create reduced resolution image thumbnail. While the computation time needed to accomplish this dependends [sic]on the processor and algorithms employed, it is clear a single computer cannot digest 10-30 million in a reasonable amount of time. This hurdle cleared using our multi-processor approach. [Sclaroff, p.2, emphasis added]</p>

Sclaroff: ImageRover

	causing said thumbnail generator to shrink said rendered images of said retrieved web pages and supply them to said image server database.	<p>In the following passage, Sclaroff discloses a thumbnail generator (the digestion module) that shrinks the rendered images and supplies them to an image server database (the image index database):</p> <p>“As shown in the Figure 1, robots can contain collection modules, digestion modules, and a local database. The collection modules recursively parse and traverse WWW documents, gathering images. The digestion modules then process these images to extract needed image indexing information and to compute a reduced resolution thumbnail image. The robots are dispatched and coordinated via a separate coordination layer, which also manages updates of the image index database.” [Sclaroff, p. 3, emphasis added]</p> <p>See also Figure 1 <i>supra</i> and its caption to see how these thumbnail visual images are rendered and supplied to the image server database.</p> <p>Additionally, Sclaroff provides details about this image reduction:</p> <p>“Each image digestion module processes an input stream of image URLs. Processing begins with translating the image file format (e.g., GIF, TIFF, JPEG) to the internal format, and performing color transformations. A reduced resolution image thumbnail is computed for use as an icon during search.” (p. 3)</p> <p>Additionally, Sclaroff motivates thumbnail image storage by calculating disk space requirements for ten million images:</p> <p>“The average space needed to store image thumbnail and X was 3K bytes. Based on this estimate, 28GB are needed to store an index for ten million images.” [Sclaroff, p. 5]</p>
17.	A method according to claim 16 also comprising deleting executable content from said	Sclaroff describes how the collection module parses said retrieved web pages to gather only images, thus deleting executable content, as a precursor to producing thumbnail images:

Sclaroff: ImageRover

	retrieved web pages.	“As shown in the Figure 1, robots can contain collection modules, digestion modules, and a local database. The collection modules recursively parse and traverse WWW documents, gathering images. The digestion modules then process these images to extract needed image indexing information and to compute a reduced resolution thumbnail image.”[Sclaroff, p. 3, emphasis added]
33.	A system for generating an image server database of thumbnail visual images of web pages, the system comprising:	See Claim 16 analysis, <i>supra</i> , which is hereby incorporated by reference
	a multiplicity of downloaders, each receiving at least one URL from a list of URLs corresponding to said web pages, the thumbnail visual images of which it is desired to supply to said image server database, and simultaneously retrieving from the Internet web pages and embedded objects corresponding to said at least one URL; and	
	a multiplicity of downloaders, each receiving at least one URL from a list of URLs corresponding to said web pages, the thumbnail visual images of which it is desired to supply to said image server database, and simultaneously retrieving from the Internet web pages and embedded objects corresponding to said at	

Sclaroff: ImageRover

	least one URL; and	
	at least one thumbnail generator operative to render the web pages, shrink said rendered images of the web pages and supply said rendered images to said image server database.	
34.	A system according to claim 33 and wherein said multiplicity of downloaders are operative to delete executable content from the web pages.	See Claim 17 analysis, <i>supra</i> , which is hereby incorporated by reference

EXHIBIT G13

US Patent 6,594,697 (Praitis)

**The following claims are invalid as being obvious under Praitis in view of
Brown, or Kraft, or Schmid, or Kopetzky**

US Patent 6,594,697. *Client system having error page analysis and replacement capabilities*. (Praitis, E., Berkun, S.) Filed: May 20, 1999, Issued Jul. 15, 2003. ('Praitis')

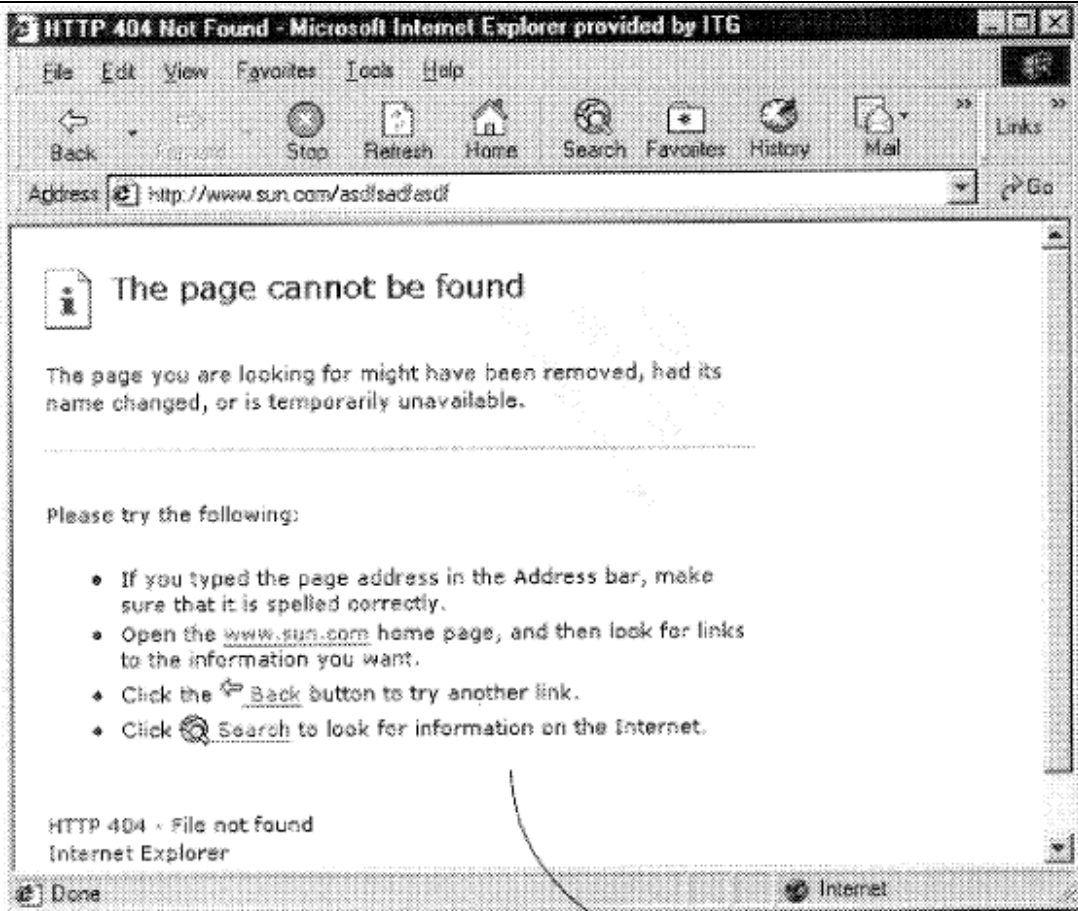



Claim #	The '904 Patent	Disclosure of Each Limitation in Praitis ¹
44.	A method according to claim 35 and wherein said visualization functionality comprises: receiving a list of hyperlinks;	<p>Praitis describes a method that creates 'friendly error pages' for end users, where the method substitutes a more informative 'friendly' page for the standard error page .</p> <p>Praitis discloses generating a substitution page that includes a link to the "home page related to the web server that was initially accessed", i.e., the domain name. As shown in [Praitis, Figure 9] below, the new page would then include a message telling the end user that they could open that 'home page' as specified by the domain name, and then look for links to the information they wanted. Finding this 'home page' involves parsing the URL, which one skilled in the art could do by trivially applying standard string or URL manipulation techniques. As Praitis discloses:</p> <p>"Creating the friendly error pages requires that a script in HTML be constructed in the usual manner. The HTML script incorporates text that is displayed on the screen and provides information related to the error that occurred and potential causes and probable techniques that may be used to avoid the error in the future. These HTML documents also incorporate hyperlinks that can be activated to navigate to safe URL. The browser assumes that the home page related to the web server that was initially accessed is a safe page and thus most error message pages incorporate a hyper link to the home page.</p> <p>In order to create the hyperlink to the related home page, the HTML document</p>
	receiving a list of hyperlinks;	
	splitting a URL of each hyperlink into URL components including at least a path component and a host component;	
	trimming a path component based on the consideration of finding the most representative image of a given web page;	
	and constructing a new URL including a trimmed path	

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

US Patent 6,594,697 (Praitis)

	component.	<p>comprises scripted code that parses the requested URL. In parsing the URL, the code searches for a first predetermined delimiter and disregards all information before the delimiter. Next, the code stores each element of the URL up to a second delimiter and disregards all information following the second delimiter. The information between the two delimiters is used to construct a URL which becomes the requested page should the hyperlink become activated. The constructed URL comprises all necessary information such as the proper protocol, delimiters, etc.” [Praitis, Col. 13:22]</p> <p>Figure 9 from Praitis below shows an example. In this figure, the original URL <code>http://www.sun.com/asd/sad/asdf</code> is in the title bar. This URL is then parsed and reassembled to construct the home page URL <code>“www.sun.com”</code> in the main browser window. Praitis discloses this when he says “displaying the error URL in the address bar 75 (FIG. 2) allows the user to immediately view the syntax and spelling of the URL entered which caused the error.” [Praitis, col 14:38].</p>
--	------------	--

US Patent 6,594,697 (Praitis)

		 <p>HTTP 404 Not Found - Microsoft Internet Explorer provided by ITG</p> <p>File Edit View Favorites Tools Help</p> <p>Back Forward Stop Refresh Home Search Favorites History Mail Links</p> <p>Address http://www.sun.com/asdfsad/asdf Go</p> <p> The page cannot be found</p> <p>The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.</p> <p>Please try the following:</p> <ul style="list-style-type: none"> • If you typed the page address in the Address bar, make sure that it is spelled correctly. • Open the www.sun.com home page, and then look for links to the information you want. • Click the  Back button to try another link. • Click  Search to look for information on the Internet. <p>HTTP 404 - File not found Internet Explorer</p> <p>Done Internet</p> <p>304</p>
55.	A system according to claim 46 and wherein said visualization functionality comprises	<i>See Claim 44 supra, which is hereby incorporated by reference.</i>
	receiving a list of hyperlinks;	

US Patent 6,594,697 (Praitis)

	splitting a URL of each hyperlink into URL components including at least a path component and a host component;	
	trimming a path component based on the consideration of finding the most representative image of a given web page; and	
	constructing a new URL including a trimmed path component.	

EXHIBIT G14

DoubleClick Ad Server

**The following claims are invalid as being obvious under DoubleClick in view of
Brown, or Kraft, or Schmid, or Kopetzky**

To my understanding, the materials below all describe the behavior of the DoubleClick Ad Server ('DoubleClick').

1. US Patent 5,948,061. *Method of delivery, targeting, and measuring advertising over networks*. (Merriman; Dwight Allen, O'Connor; Kevin Joseph; Assigned to DoubleClick) Filed Oct. 29, 1996, Issued Sep 7, 1999. ('Merriman')
2. DoubleClick web site. ('DoubleClick')
 - Intro. Marked with the date 06.25.1998. Marked IACGIR0002472 to 4.
 - Technical Specs. Marked with the date 02.05.1998. Marked IACGIR0002475
 - Site Modifications. Marked with the date 02.05.1998. IACGIR0002476 to 9.
 - On Delivery. Marked with the date 02.05.1998. IACGIR0002480.
3. Langheinrich, M., Atsuyoshi Nakamura, Naoki Abe, Tomonari Kamba, Yoshiyuki Koseki: *Unintrusive Customization Techniques for Web Advertising*. Proc. 8th International World Wide Web Conference. pp. 181-194, Toronto, May 1999. ('Langheinreich')

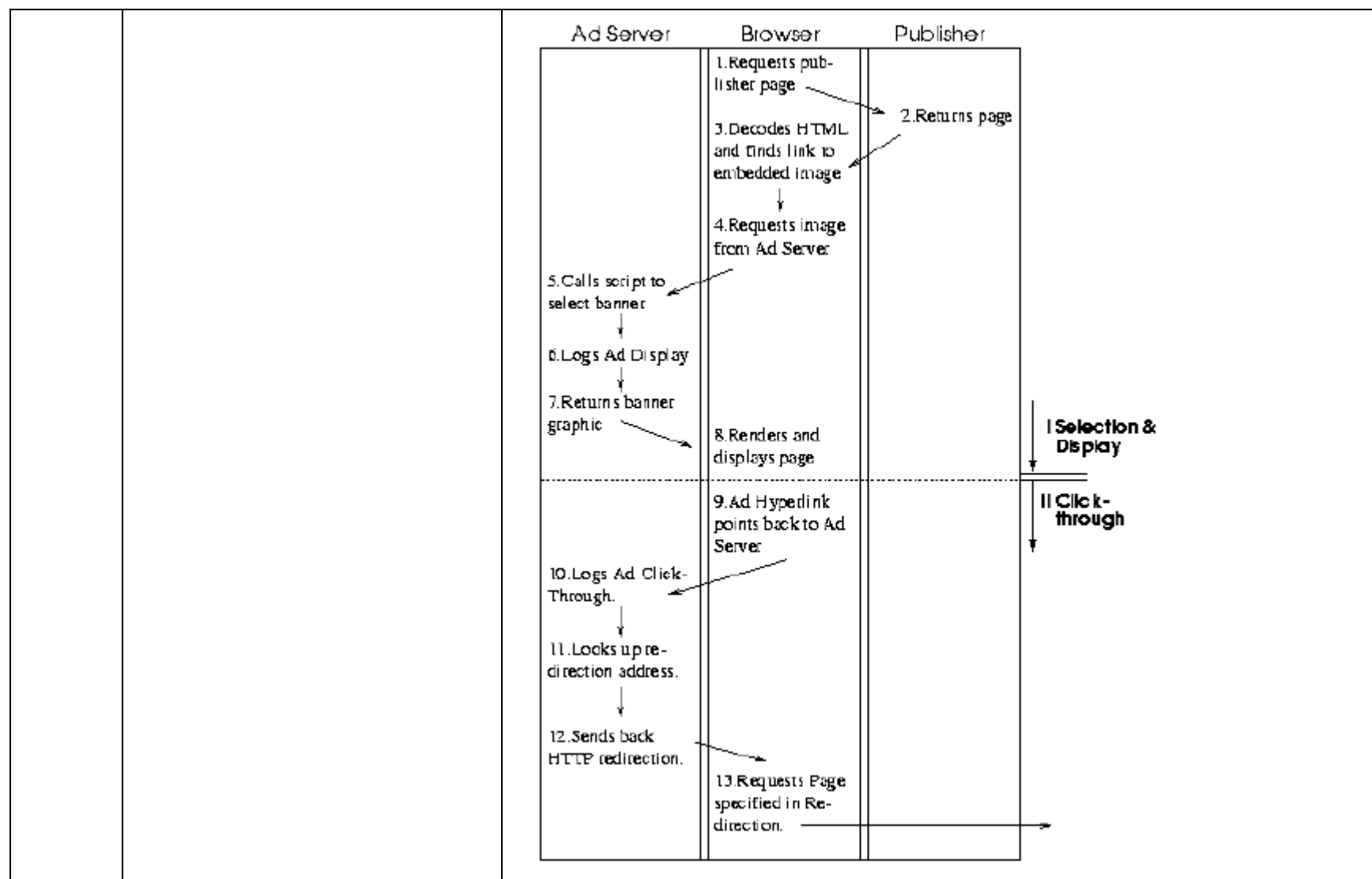
Claim #	The '904 Patent	Disclosure of Each Limitation in DoubleClick ¹
35.	A method for presenting Internet information to a user comprising: providing to a user a visual image of a web page containing at least	The idea of employing an image server that stores and provides said thumbnail visual image, where the image server is separated from a web server, was well established in other contexts before the provisional application that became the '904 patent was filed.

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

DoubleClick Ad Server

	<p>one hyperlink;</p> <p>and at least partially concurrently providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image,</p> <p>said providing a thumbnail visual image comprising employing a web browser which interfaces via the Internet with a web server, separated from said image server, including visualization functionality, said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.</p>	<p>The most obvious example is in the web banner advertisement context. By the mid-1990s, banner advertisement providers, including the well-known DoubleClick, employed their own image servers to serve small images to a variety of client websites. Customers of the banner advertisement providers hosted their web sites on their own, separate, web servers.</p> <p>The then-current state of the art was well-described in a paper by Langheinrich presented in May of 1999:</p>
--	--	--

DoubleClick Ad Server



DoubleClick Ad Server

		<p>Figure 2: Online Advertisement - Control & Data Flow. Embedded HTML images (step 3) and HTTP redirection (step 12) make it possible to separate page content and advertisement. After obtaining the page content from the publisher's Web server, the user's browser will request a dynamically selected advertisement from the Ad Server (part I). Once the user clicks on the advertisement (part II), the Ad Server will redirect the user to the appropriate site.</p> <p>[Langheinrich at 2.1]</p> <p>"The basic process is outlined in figure 2: the Web page of an online service (the "publisher") contains a link to a banner advertisement. Although the content of the original Web page (step 2) stays the same, the Ad Server will potentially select different banner images for subsequent advertisement requests (steps 5-7)."</p> <p>[Langheinrich at 2.1]</p> <p>This system for providing banner advertisements via a separate image server was well-known in the art, as exemplified by DoubleClick.</p> <p>An Internet Archive version of the DoubleClick website from early 1998 explains: "How Does DART Do It? When a user visits a DART-enabled Web site, the selected Web page is delivered from that Web site to the User, and the User's browser loads the requested page. Embedded in the page are Image Tags that link the browser to the closest DoubleClick DART ad server. The User's browser initiates the HTTP Get request from the server, establishing a connection between the two, and a graphic file is requested from DART to fill the ad banner space on the Web page being loaded on the User's screen."</p> <p>DoubleClick emphasized the advantage of a centralized image server, separate from the customer's web server:</p> <p>"DoubleClick helps sites avoid the costly investment of purchasing and maintaining</p>
--	--	--

DoubleClick Ad Server

		<p>ad servers by using a centralized ad serving solution. Sites simply include HTML tags on their pages and DoubleClick matches an appropriate targeted banner, in less than 20 milliseconds. . . . As a DART client, you will benefit from these innovations as we develop them. Because all ad serving is done from DoubleClick's central servers, there is no need to upgrade or purchase new equipment." [DoubleClick]s</p> <p>DoubleClick reports additional advantages:</p> <ol style="list-style-type: none"> 1. "Ad Delivery. DoubleClick helps sites avoid the costly investment of purchasing and maintaining ad servers by using a centralized ad serving solution. Sites simply include HTML tags on their pages and DoubleClick matches an appropriate targeted banner, in less than 20 milliseconds." [DoubleClick-1] 2. "This centralized solution allows sites to take advantage of our ability to aggregate and target based on user data. Soon, this will allow us to deliver demographically targeted ads." [DoubleClick-1]. "DoubleClick's systems are extremely redundant and have operated at 99.9% uptime since 1/97" [DoubleClick] <p>DoubleClick provided pseudo-code to demonstrate how a web site would use their product. Note that this pseudo-code "is operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages," as required by Claims [X, Y, and Z] of the '904 patent:</p> <p>"This is how each site should set up their code:</p> <pre><html> <title> The following "base" tag will be needed for Netscape 4 users. Each layer will have to know where to get it's [sic] paths from. <base href="http://www.yoursite.com"> <body> Here is where you will code up the content of your site, just as you normally would. When it comes time to put the banner on the page, the following code takes over. <nolayer> In here, you will be putting in the "IFrame" tags that will allow Internet Explorer Users to launch an "IFrame" and give that user the ability to see an enhanced banner.</pre>
--	--	--

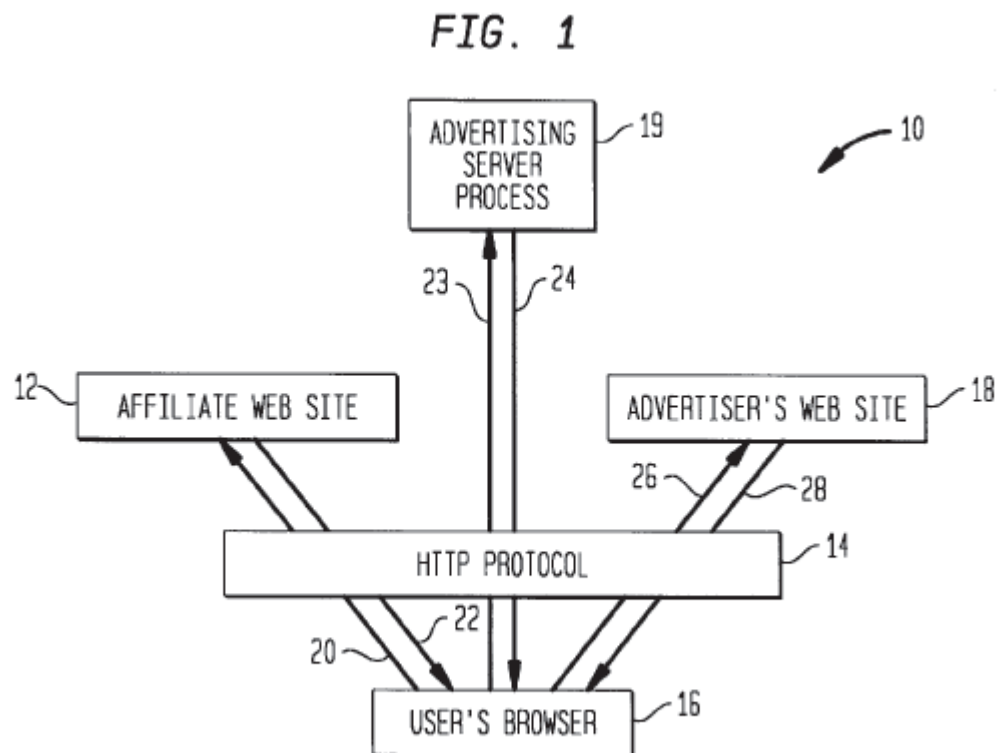
DoubleClick Ad Server

		<p><iframe src="http://ad.doubleclick.net/adi/unique_url_goes_here" frameborder=0 vspace=0 hspace=0 marginwidth=0 marginheight=0 width=468 height=60> In here, regular DART ad tags will be inserted to make sure that all browsers are covered. Now we have to end the entire "IFrame" section. </iframe> The "NoLayer" section is over now. We've laid down the banner for users with any browser except Netscape 4.0 and above. </nolayer> Now the rest of the page gets coded up. At the very bottom of the page/code, we will put the "Layer" tag that will call the banner, in a layer, from one of our ad servers. A "Layer" is basically a transparency, so the only thing visible is the text and/or images embeded in the layer. The DART ad servers work to output the banner and the hyper reference in the layer. Now, to be able to put the banner-layer in the perfect position over the content-layer, you will have to do some trial-and-error type guessing. A set of parameters in a "Layer" tag allow you to start the layer a specified number of pixels from the top, left, bottom, or right sides of the screen. If you have nothing above the banner and you just want the banner to appear at the top of the page, you're "Layer" tag would look like this: <layer src="http://ad.doubleclick.net/adl/unique_url_goes_here"> If you had a logo, let's say, at the top of your screen, and you were going to put the banner a specified distance down from the top of the screen, your "Layer" tag might look like this: <layer src="http://ad.doubleclick.net/adl/unique_url_goes_here" top=80>”</p> <p>This example HTML code demonstrates that the images are supplied from the DoubleClick image servers: “” while the underlying web page is generated by the customer’s own web server.</p>
--	--	--

DoubleClick Ad Server

		<p>Merriman, filed on October 29, 1996, and assigned to Double Click, Inc., discloses a similar system:</p> <p>“To effect such a capability, an advertising server process is provided as a node on the network. The various advertisements are stored on the network of the server and preferably on the server. When, for example, a user using a web browser accesses a web page that is affiliated with the advertising server process, the affiliated page’s encoding includes an embedded reference to an object provided by the advertising server process. That causes the user’s browser to contact the advertising server process to provide the advertising image or information that will appear on the accessed web page as displayed by the user’s browser.” [Merriman at col. 2:15-25]</p> <p>“The basic operation of the system is as follows in the preferred embodiment. When a user browsing on the Internet accesses an affiliate’s web site 12, the user’s browser generates an HTTP message 20 to get the information for the desired web page. The affiliate’s web site in response to the message 20 transmits one or more messages back 22 containing the information to be displayed by the user’s browser. In addition, an advertising server process 19 will provide additional information comprising one or more objects such as banner advertisements to be displayed with the information provided from the affiliate web site. Normally, the computers supporting the browser, the affiliate web site and the advertising server process will be at entirely different nodes on the Internet. Upon clicking through or otherwise selecting the advertisement object, which may be an image such as an advertisement banner, an icon, or a video or an audio clip, the browser ends up being connected to the advertiser’s server or web site 18 for that advertisement object.” [Merriman at col. 3:5-23]</p> <p>See also, Figure 1:</p>
--	--	---

DoubleClick Ad Server



“In FIG. 1, a user operates a web browser, such as Netscape or Microsoft Internet Explorer, on a computer or PDA or other Internet capable device 16 to generate through the hypertext transfer protocol (HTTP) 14 a request 20 to any one of preferably a plurality of affiliate web sites 12. The affiliate web site sends one or more messages back 22 using the same protocol. Those messages 22 preferably contain all of the information available at the particular web site 12 for the requested page to be displayed by the user’s browser 16 except for one or more advertising objects such as banner advertisements. These objects preferably do not reside on the affiliate’s web server. Instead, the affiliate’s web server sends back a link including

DoubleClick Ad Server

		<p>an IP address for a node running an advertiser server process 19 as well as information about the page on which the advertisement will be displayed. The link by way of example may be a hypertext markup language (HTML) tag, referring to, for example, and inline image such as a banner. The user's browser 16 then transmits a message 23 using the received IP address to access such an object indicated by the HTML tag from the advertisement server 19. . . . <u>Upon receiving the request in the message 23, the advertising server process 19 determines which advertisement or other object to provide to user's browser and transmits the messages 24 containing the object such as a banner advertisement to the user's browser 16 using the HTTP protocol.</u> . . . That advertisement object is then displayed on the image created by the web user's browser as a composite of the received affiliate's web page plus the object transmitted back by the advertisement web server." [Merriman at col. 3:24-63] (emphasis added)</p> <p>"Each advertisement along with a table of the targeting profile criteria for the advertisement and other data regarding the advertisement currently available is stored in a database such as shown in FIG. 3B. The actual advertising object, which may be a banner image in a GIF or JPEG file format, an icon for an audio or video clip or some other object is kept as part of the advertising server process." [Merriman at col. 5:64-6:3]</p>
46.	A system for presenting Internet information to a user comprising:	<i>See Claim 35 supra, which is hereby incorporated by reference.</i>
	first functionality providing to a user a visual image of a web page containing at least one hyperlink; and	
	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of another web page of at	

DoubleClick Ad Server

	least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image, said second functionality comprising third functionality employing a web browser which interfaces via the Internet with a web server, separated from said image server, including visualization functionality,	
	said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.	

EXHIBIT G15

US Patent 6,108,703 (Leighton)

**The following claims are invalid as being obvious under US Patent 6,108,703 (Leighton) in view of
Brown, or Kraft, or Schmid, or Kopetzky**

US Patent 6,108,703. *Global Hosting System*. (Leighton, F. and Lewin, D.) Filed May 19, 1999, Issued Aug. 22, 2000. ('Leighton')

Claim #	The '904 Patent	Disclosure of Each Limitation in Leighton ¹
35.	<p>A method for presenting Internet information to a user comprising:</p> <p>providing to a user a visual image of a web page containing at least one hyperlink;</p> <p>and at least partially concurrently providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image,</p> <p>said providing a thumbnail visual image comprising employing a web browser which interfaces via the Internet with a web server,</p>	<p>Leighton discloses how a content server separate from the web page server can provide HTML page objects, including images, to clients such as browsers. An image server thus is just a type of content server. Leighton discloses how the base HTML document is served from a content provider, while a different, separate server serves the HTML objects:</p> <p>“The inventive framework comprises a set of servers operating in a distributed manner. The actual content to be served is preferably supported on a set of hosting servers (sometimes referred to as ghost servers). This content comprises HTML page objects that, conventionally, are served from a Content Provider site. In accordance with the invention, however, a base HTML document portion of a Web page is served from the Content Provider's site while one or more embedded objects for the page are served from the hosting servers, preferably, those hosting servers near the client machine. By serving the base HTML document from the Content Provider's site, the Content Provider maintains control over the content. . [Leighton, abstract]”</p> <p>The content server they describe covers any embedded content servable as separate objects. This includes, of course, images, which Leighton recognizes:</p>

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

US Patent 6,108,703 (Leighton)

	<p>separated from said image server, including visualization functionality, said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.</p>	<p>“As seen in FIG. 2, a typical Web page comprises a markup language (e.g. HTML) master or base document 28, and many embedded objects (e.g., images, audio, video, or the like) 30. Thus, in a typical page, twenty or more embedded images or objects are quite common. Each of these images is an independent object in the Web, retrieved (or validated for change) separately. The common behavior of a Web client, therefore, is to fetch the base HTML document, and then immediately fetch the embedded objects, which are typically (but not always) located on the same server. According to the present invention, preferably the markup language base document 28 is served from the Web server (i.e., the Content Provider site) whereas a given number (or perhaps all) of the embedded objects are served from other servers. As will be seen, preferably a given embedded object is served from a server (other than the Web server itself) that is close to the client machine, that is not overloaded, and that is most likely to already have a current version of the required file. [Leighton, 5:23]”</p> <p>Leighton offer various advantages to content servers, which would also apply to image servers. A few are listed below:</p> <p>“It is a general object of the present invention to provide a computer network comprising a large number of widely deployed Internet servers that form an organic, massively fault-tolerant infrastructure designed to serve Web content efficiently, effectively, and reliably to end users.</p> <p>Another more general object of the present invention is to provide a fundamentally new and better method to distribute Web-based content. The inventive architecture provides a method for intelligently routing and replicating content over a large network of distributed servers, preferably with no centralized control.</p> <p>Another object of the present invention is to provide a network architecture that moves content close to the user. The inventive architecture allows Web sites to develop large audiences without worrying about building a massive infrastructure to handle the associated traffic.</p>
--	---	---

US Patent 6,108,703 (Leighton)

		<p>Still another object of the present invention is to provide a fault-tolerant network for distributing Web content. The network architecture is used to speed-up the delivery of richer Web pages, and it allows Content Providers with large audiences to serve them reliably and economically, preferably from servers located close to end users.</p> <p>A further feature of the present invention is the ability to distribute and manage content over a large network without disrupting the Content Provider's direct relationship with the end user.</p> <p>Yet another feature of the present invention is to provide a distributed scalable infrastructure for the Internet that shifts the burden of Web content distribution from the Content Provider to a network of preferably hundreds of hosting servers deployed, for example, on a global basis.</p> <p>In general, the present invention is a network architecture that supports hosting on a truly global scale. The inventive framework allows a Content Provider to replicate its most popular content at an unlimited number of points throughout the world. As an additional feature, the actual content that is replicated at any one geographic location is specifically tailored to viewers in that location. Moreover, content is automatically sent to the location where it is requested, without any effort or overhead on the part of a Content Provider.” [Leighton, 2:40]</p> <p>Leighton discloses a visualization functionality that modifies the URLs embedded in an HTML document to point instead to URLs of the content server which will supply those objects instead. Leighton also illustrates this process using an example where an image URL is modified:</p> <p>“In particular, it is assumed that a user of a client machine in Boston requests a Content Provider Web page normally hosted in Atlanta. For illustrative purposes, It is assumed that the Content Provider is using the global hosting architecture within a network, which may be global, international, national, regional, local or private. FIG. 5 shows the various components of the system and how the request from the client is processed. This operation is not to be taken by way of limitation, as will be explained.</p>
--	--	--

US Patent 6,108,703 (Leighton)

		<p>Step 1: The browser sends a request to the Provider's Web site (Item 1). The Content Provider site in Atlanta receives the request in the same way that it does as if the global hosting framework were not being implemented. The difference is in what is returned by the Provider site. Instead of returning the usual page, according to the invention, the Web site returns a page with embedded object URLs that are modified according to the method illustrated in the flowchart of FIG. 4. As previously described, the URLs preferably are changed as follows:</p> <p>Assume that there are 100,000 virtual ghost servers, even though there may only be a relatively small number (e.g., 100) physically present on the network. These virtual ghost servers or virtual ghosts are identified by the hostname: ghostxxxxx.ghosting.com, where xxxxx is replaced by a number between 0 and 99,999. After the Content Provider Web site is updated with new information, a script executing on the Content Provider site is run that rewrites the embedded URLs. Preferably, the embedded URLs names are hashed into numbers between 0 and 99,999, although this range is not a limitation of the present invention. An embedded URL is then switched to reference the virtual ghost with that number. For example, the following is an embedded URL from the Provider's site:</p> <p></p> <p>If the serial number for the object referred to by this URL is the number 1467, then preferably the URL is rewritten to read:</p> <p>.”</p>
46.	A system for presenting Internet information to a user comprising:	<i>See Claim 35 supra, which is hereby incorporated by reference.</i>
	first functionality providing to a user a visual image of a web page	

US Patent 6,108,703 (Leighton)

	containing at least one hyperlink; and	
	second functionality operative at least partially concurrently with said first functionality for providing a thumbnail visual image of another web page of at least one web site which is represented by said at least one hyperlink via the Internet by employing an image server that stores and provides said thumbnail visual image, said second functionality comprising third functionality employing a web browser which interfaces via the Internet with a web server, separated from said image server, including visualization functionality,	
	said visualization functionality being operative to embed commands to the web browser to download, via said image server, thumbnail visual images of web pages which represent hyperlinks contained in the web page and to provide to a user, via the web browser, an annotated web page.	

EXHIBIT G16

US Patent 5,761,436 (Nielsen)

**The following claims are invalid as being obvious under Nielsen in view of
Brown, or Kraft, or Schmid, or Kopetzky**

US Patent 5,761,436. *Method and apparatus for combining truncated hyperlinks to form a hyperlink aggregate.* (Jakob Nielsen)
Filed Jul 1, 1996. Issued Jun 2, 1998. ('Nielsen')

Claim #	The '904 Patent	Disclosure of Each Limitation in Nielsen¹
44.	A method according to claim 35 and wherein said visualization functionality comprises:	<p>Nielsen discloses a method “for maintaining a history of accessed hyperlinks used to access hypernodes” [Nielsen, Col 1:10] where the invention “provides a user of a hypertext system with an enhanced history presentation that allows the user to more quickly find and reference previously vided hypernodes.” [Nielsen, Col. 2:32]</p> <p>Nielsen discloses a visualization functionality that receives a list of hyperlinks, splits the URL into components, trims the path to find the most representative link, and constructs a new URL including the trimmed path component. This visualization functionality is illustrated in Figure 9 below [Nielsen], where the functionality is described as:</p> <p>“the process used to extract truncated hyperlinks from existing data history records in accordance with a preferred embodiment” [Nielsen, Col. 3:55]</p>
	receiving a list of hyperlinks;	
	splitting a URL of each hyperlink into URL components including at least a path component and a host component;	
	trimming a path component based on the consideration of finding the most representative image of a given web page;	
	and constructing a new URL including a trimmed path component.	

¹ I reserve the right to revise this report and charts attached thereto concerning the invalidity of the asserted claims depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues. It is also my understanding that certain discovery has yet to be conducted or completed in this matter and I further reserve the right to supplement my report should additional information become available.

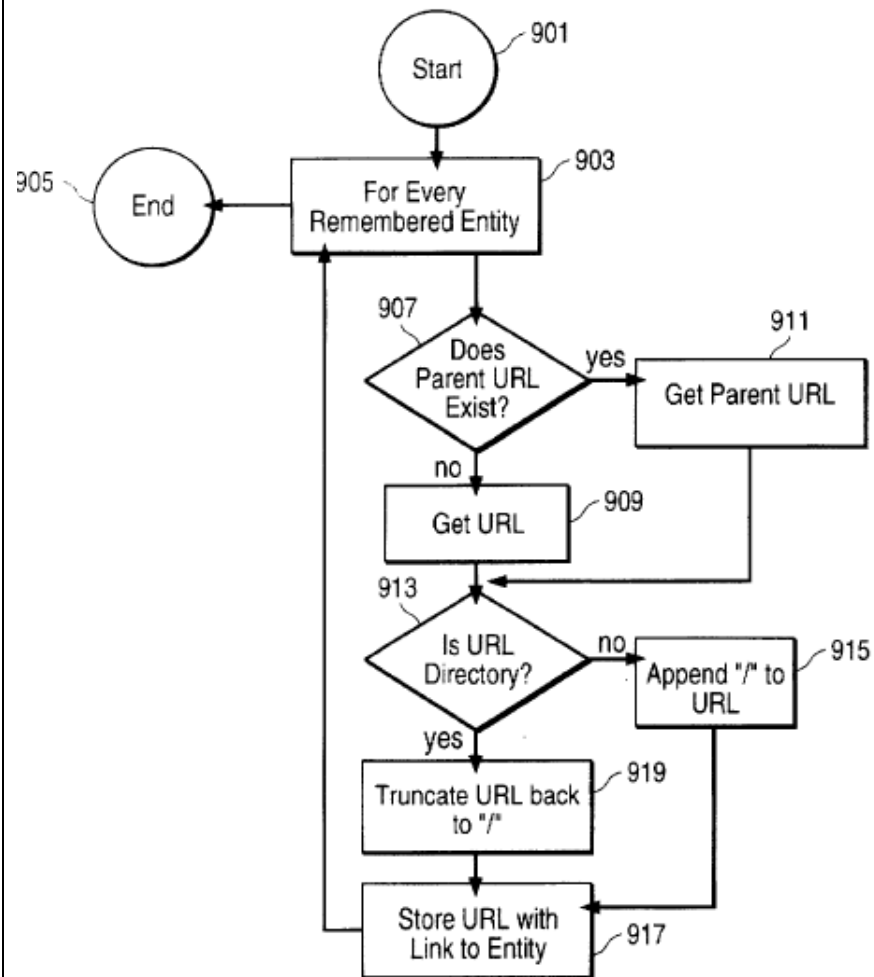


FIG. 9

	<p><i>Nielsen verifies that the URL protocol contains well-specified components:</i> “Briefly, the URL contains a protocol specification and a path specification. The protocol specification notifies the browser of what protocol to use when accessing the remote server. The path specification is generally a hierarchical path that specifies a data server followed by a hypernode (such as a Web Page) that actually provides the information for the browser.” [Nielsen, Col. 1:43-49]</p> <p><i>Nielsen discloses receiving a list of hyperlinks:</i> “The method first creates a plurality of truncated hyperlink references by determining a truncated hyperlink for each of the plurality of hyperlinks.” [Nielsen, Col. 2:35, emphasis added]</p> <p><i>Nielsen discloses that these hyperlinks are to be truncated and combined:</i> “The method first creates a plurality of truncated hyperlink references by determining a truncated hyperlink for each of the plurality of hyperlinks. Next, the method matches a first truncated hyperlink reference to a second truncated hyperlink. The first and second truncated hyperlinks respectively reference a first and second information. Next, the method combines the first and second truncated hyperlink reference to form a hyperlink aggregate. This hyperlink aggregate also having aggregate information. Finally, the method presents the aggregate information in place of presenting the first and second information.” [Nielsen, Col. 2:35, emphasis added]</p> <p><i>One embodiment describes how a node is trimmed from the path:</i> “The step 717 creates a hyperlink sorting data record 690 and extracts a copy of the hyperlink specification from each entity. The step 717 also links the hyperlink sorting data record 690 back to its source entity by storing a pointer to the entity in the "Pointer to Entity" field 693 of the record. If the lowest level of the hierarchy of the hyperlink explicitly specifies a hypernode, the step 717 also removes the hypernode from the hyperlink specification. This process is described in FIG. 9 and results in a truncated hyperlink specifying a hierarchical reference that is one step removed from a specific hypernode. (That is, in the WWW instance, "http://www.sun.com/netra-nfs/features.html" is truncated to</p>
--	---

	<p>"http://www.sun.com/netra-nfs/")." [Nielsen, Col. 10:50, emphasis added]</p> <p><i>Another embodiment describes how the path component is trimmed based on the consideration of finding and constructing a representative URL from a set of web pages, i.e., one that represents the common hyperlink hierarchy:</i></p> <p>"In another preferred embodiment, this step 717 first locates all the entities having a common META parent URL and aggregates these entities and saves the aggregate in the new history list. This allows a parent aggregation that includes component hypernodes having completely different hyperlinks. In both of these preferred embodiments, the process continues to a sort hyperlink step 719.</p> <p>At the sort hyperlink step 719, the truncated hyperlinks are sorted according to their hierarchical structure. This sorting step results in identical matching truncated hyperlinks being grouped together. Next at a step 721 the sorted truncated hyperlinks are examined to locate singleton truncated hyperlinks. The entity pointed to by the "Pointer to Entity" field 693 of each of the singleton truncated hyperlinks is moved to the new history list and the corresponding sorting data record 690 is deleted. At this point the truncated hyperlinks consist of multiple occurrences of matching truncated hyperlinks grouped together. These grouped truncated hyperlinks have common hyperlink hierarchy. These groups of truncated hyperlinks define which entities are aggregated. At a step 723 the process builds aggregates by combining the truncated hyperlinks from these groups. This process is further described below for FIG. 10. At this point, the current entities have been saved in the new history list, the entities that can not be aggregated have been saved in the new history list, and now at a step 725 the newly created aggregates are saved in the new history list. At a step 727, the old history list is replaced by the new history list and the process completes through a terminal 729." [Nielsen, Col. 10:63-11:25, emphasis added]</p> <p><i>The method finds a representation of the URL that will ultimately be presented to the</i></p>
--	--

US Patent 5,761,436 (Nielsen)

		<p><i>user:</i></p> <p>“...Next at a decision block 1011 the group is checked to determine whether any of the entities comprising the elements of the group have a parent hypernode. If the decision block 1011 is satisfied (by detecting a non-NIL entity in the "Pointer to META Parent URL" field 531), a step 1013 uses the parent's hypernode title for the title of the aggregate. The step 1013 also detects (not shown) if there are multiple non-aggregate history data structures in the aggregate that have different parent hyperlinks and if so, simply saves the first entity with such a parent hyperlink in the new history list and removes the offending entity from the aggregate. Next the process continues to a step 1015 that advances to the next truncated hyperlink group and continues processing at the iteration step 1003. However if at the decision block 1011, the group did not include an entity with a parent URL, the directory URL is placed in the "Page Title" field 517. Next the process continues to a step 1015 that advances to the next truncated hyperlink group and continues processing at the iteration step 1003.” [Nielsen, Col. 12:45-64]</p>
55.	A system according to claim 46 and wherein said visualization functionality comprises	<i>See Claim 44 supra, which is hereby incorporated by reference.</i>
	receiving a list of hyperlinks;	
	splitting a URL of each hyperlink into URL components including at least a path component and a host component;	
	trimming a path component based on the consideration of finding the most representative image of a given web page; and	

US Patent 5,761,436 (Nielsen)

	constructing a new URL including a trimmed path component.	
--	--	--